

REMARKS

In the Official Action mailed on **18 April 2007**, the Examiner reviewed claims 1-3, 5-16, 18-29 and 31-39. Claims 1-3, 5-16, 18-29, and 31-39 were rejected under 35 U.S.C. § 103(a) as being anticipated by Wu et al. (US Pub. No. 2005/0033726 hereinafter “Wu”), and further in view of Gary Willman Policastro (US Pub. No. 2003/0018605 hereinafter “Policastro”).

Objection to Affidavit C.F.R. 1.131

In the office action of 18 April 2007, Examiner avers that the evidence submitted in support of the Affidavit – 37 C.F.R. 1.131 was insufficient to establish a reduction to practice of the invention prior to the effective date of the cited prior art references. In the phone conversation with the Examiner of 4 May 2007, Examiner suggested that this rejection could be overcome by (1) establishing an actual reduction-to-practice data prior to the effective date of the reference, and (2) mapping the product to the elements of the independent claims. In response, applicant provides the following evidentiary support to establish both the reduction-to-practice date and the mapping between the produce the claim limitations.

Actual Reduction to Practice

As declared in the 37 C.F.R. § 1.131 declaration executed on 6 February 2007, embodiments of the present invention were implemented in Oracle Warehouse Builder 9.0.4 (hereinafter the “product”), which was commercially available in March 2003. The effective date of the main reference, Wu, is 19 May 2003. Therefore, the actual reduction-to-practice of embodiments of the present invention predates Wu. The following third-party news releases show the date when this product became commercially available (the printouts are attached to this paper).

http://www.infoworld.com/article/03/03/12/HNwarehousebuilder_1.html

<http://www.computerworld.com/databasetopics/data/software/story/0,10801,79326,00.html>

<http://www.internetnews.com/ent-news/article.php/2108471>

Mapping Between Product the Claim Elements

Applicant avers that the independent claims are disclosed in the Oracle Warehouse Builder 9.0.4 manual, which can be found at:

http://download-west.oracle.com/docs/html/B10657_01/toc.htm

Specifically, claim 1 includes the following operations:

- 1) creating metadata, under the control of a user, for use in the data warehouse system;
- 2) moving the metadata, under the control of a super user, into and out of a collection, thereby securing the metadata;
- 3) assigning, under the control of the super user, a data steward for the collection; and
- 4) manipulating, under the control of the data steward, the metadata in the collection, wherein manipulating the metadata involves securing the metadata and performing administration operations on the collection, wherein only the data steward can manipulate the metadata in the collection, and wherein manipulating the metadata includes editing and deleting the metadata.

Operation 1) is disclosed in chapter 14, on page 19, “Add new metadata only (Create Mode)...,” and in chapter 18 on page 1, “You create and manage UDPs using the Oracle MetaBase (OMB) Scripting Language.”

Operation 2) is disclosed in chapter 14, on page 1, “The Metadata Loader (MDL) enables you to populate a new repository as well as transfer...,” and on page 1, “You can import and export metadata...”, in chapter 17, on page 10, “Click register an OWB Repository on the Warehouse Builder Administration home page...,” and in chapter 18, on page 5, “The primary method for propagating changes...”

Operation 3) is disclosed in chapter 17, on page 11, Table 17-2 Repository Management, on page 19, “... one of these Warehouse builder pre-defined roles... The Administrator can assign these roles...,” on page 21, “The preferences you can save include... Access rights associated with roles, repositories...”, and in chapter 18, on page 6, “Multiple identifiable Warehouse Builder users can access the same central repository schema when they are registered by the repository owner,” and on page 7, “Register repository users.”

Operation 4) is disclosed in chapter 18, on page 11, “To control users who can invoke...,” on page 11, “You can also use it to control which use can invoke this operation,” on page 11, “Definition of constants for all basic operations,” on page 11, “You can also use it to control which use can invoke this service operation,” on page 12, “You can also use it to implement your access control...”

Chapters 14, 17, and 19 of the produce manual are attached with this paper.

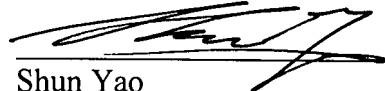
Applicant avers the above cited references disclose the teachings of independent claim 1, and similar the teachings of independent claims 14 and 27. Hence, Applicant avers that the present invention pre-dates the prior art and thus, the claims of the instant application are in condition for allowance.

CONCLUSION

It is submitted that the instant application is presently in condition for allowance. Such action is respectfully requested.

Respectfully submitted,

By




Shun Yao

Registration No. 59,242


Date: 15 May 2007

Shun Yao
PARK, VAUGHAN & FLEMING LLP
2820 Fifth Street
Davis, CA 95618-7759
Tel: (530) 759-1667
FAX: (530) 759-1665
Email: shun@parklegal.com



Microsoft Virtual Machine Manager
Beta 2 is Here!

Download Beta 2 for free
Enter to win an Xbox 360 system!



FREE NEWSLETTERS

☐ Technology & Business Daily

Enter Email Address

» All InfoWorld Newsletters

InfoWorld





Log-in | Register

HOME ► NEWS ► TECHNOLOGIES ► BLOGS ► COLUMNS ► TEST CENTER ► AUDIO/VIDEO ► CAREERS ► IT EXEC-CONNECT ► EVENTS ► NEWSLETTERS ► RESOURCES ►

Oracle upgrades data warehouse tool

Links to database emphasized

By Paul Krill
March 12, 2003

 E-mail  Printer Friendly  Reprints  Slashdot It!

Oracle on Wednesday is shipping Oracle9i Warehouse Builder Version 9.0.4, an enterprise integration tool for managing data in business intelligence applications.

Free IT resource

Hear how top CIOs turn change into a competitive advantage.

Sponsored by HP

Free IT resource

Attend the SOA Executive Forum: Breaking SOA Bottlenecks
SOAExecForum.com/may2007

Sponsored by InfoWorld

The software helps reduce the cost and time associated with data integration by providing an ETL (extraction, transformation, and loading) tool that enables database administrators to design and build business intelligence applications, according to Oracle. Visual models and wizard-driven interfaces are featured for graphically designing end-to-end ETL processes including warehouses and intermediate storage areas.

The product is optimized for all versions of the Oracle database and takes advantage of ETL functions in the latest release of the database, Oracle9i Database Release 2. Warehouse Builder 9.0.4 also utilizes the OnLine


Analytical Processing (OLAP) engine in the new database.

Other new features include support for data extraction from SAP R/3 systems, a RunTime Audit Browser to ensure data quality while exposing audit and history information to verify job results, and a Process Flow Editor for visually building ETL transformations.

Oracle9i Warehouse Builder Version 9.0.4 is sold as part of the Oracle9i Developer Suite, which costs \$5,000 per developer. It is supported on platforms such as Linux, Sun Microsystems Solaris, and Microsoft Windows.

IMPROVE THE EFFICIENCY OF
YOUR DATA CENTER WITH
AN ARRAY OF SUN SYSTEMS
AND STORAGE.

>> FREE 60-DAY TRIAL



 E-mail  Printer Friendly  Reprints  Slashdot It!


Paul Krill is an InfoWorld editor at large.

TOP NEWS:

- » **Former Oracle VP pays \$198,000 in trading charge**
Oracle exec used confidential information on pending acquisitions to make illegal stock market profits
- » **U.S. piracy crackdown nets 50th conviction**
Ongoing DOJ investigation nabs member of Apocalypse Crew, provider of pre-release copyrighted music to the Internet
- » **Researcher: Apple TV, iTunes video "dead ends"**
Paid video downloads will give way to ad-backed streaming media and subscription model, according to Forrester analyst
- » **Microsoft patent claims hint at internal issues**
Company's claims raise questions about the business strategy behind its aggressive moves to seek licensing money from patents
- » **Passive RFID tag market to hit \$486M in 2013**
Despite current and anticipated growth, the rate falls short of vendor hopes
- » **Update: Verizon Business to acquire Cybertrust**
Verizon Business aims to become a leading provider of managed information security services

**CONTINENTAL AG
WINS BIG WITH ROCK
SOLID RELIABILITY.**

"We needed rock-solid reliability, and we weren't getting it from our legacy infrastructure," says Paul Schwefer, CIO at Continental AG.

 **STORY CONTINUES >**

FIND PRODUCTS AND COMPANIES

» COMPLETE PRODUCT GUIDE

Search Jobs **MORE OPTIONS** **careerbuilder.com**

IT Jobs **Enter City** **SEARCH**

TECHNOLOGY INDEX

- Applications
- Application Development
- Security
- Networking
- Wireless
- Platforms
- Hardware
- Data Management
- Storage
- Web Services
- Business
- Telecom
- Professional Services
- Standards

TECH WATCH

Microsoft online storage disappears overnight
Assuming you blinked this weekend, you probably missed an inadvertently short sneak peak of what could be the next big services showdown between Microsoft and Google: online storage. Microsoft's Live Drive cum Windows Live Folders -- which will


CCube targets Craigslist users for click to call
Startup VoIP social network wants to give Craigslist posts a voice. [READ MORE](#)


COLUMNISTS

Should vendors close all security holes?
(InfoWorld) - In last week's column, I argued that vendors should close all known security holes. A reader...

» [MORE COLUMNISTS](#)

MORE INFOWORLD BLOGS

Open Sources 
Rejecting the right to yesterday's business model
I was talking with a friend today and he suggested that proprietary vendors have a "right" to their business models, ...

Database Underground 
SQL Refactor is still the Bomb!
Quite a few months ago, Red-gate came out with SQLRefactor, a tool that let's you setup templates to organize your SQL ...

- Advice Line
- Database Underground
- The Deep End
- Enterprise Mac
- Geeks in Paradise
- Grid Meter
- The Gripe Line
- InfoWorld Daily
- Inside IT
- IT Troubleshooter
- ITxtreme
- Open Sources
- ProdBlog

InfoWorld WEBCASTS

E-MAIL ARCHIVE AND COMPLIANCE SOLUTIONS
Join this Webcast and learn how to accelerate archive and compliance, eliminate the need for separate storage silos without sacrificing performance, and increase data management efficiency at a lower TCO. Sponsored by NetApp.

» [Click here to view this Webcast](#)



Fundamentals of Mail Security
Net crooks' techniques have evolved with internet usage. This guide examines the results of Symantec's 2006 security surveys, evaluates spam and malware solutions, and analyzes security assessment appliances. Download a copy of this new Mail Security Sponsored Solutions Guide. Sponsored by Symantec.

» [Click here to download now](#)

BMC AppSight



Presented by CIO, CSO &
COMPUTERWORLD
Live Event: May 23, 2007
Register now >>

COMPUTERWORLD
Business Intelligence



JUMP TO More Resources

SEARCH **GO**

- Home
- News
- E-mail Newsletters
- Tech Dispenser
- + Shark Bait
- Knowledge Centers
 - + Operating Systems
 - + Networking & Internet
 - + Mobile & Wireless
 - + Security
 - + Storage
 - + Business Intelligence
 - + Servers & Data Center
 - + Hardware
 - + Software
 - + Development
 - + Careers
 - + Management
 - + Government
- + Opinion/Blogs
- Webcasts
- Podcasts
- White Papers
- + Computerworld Reports
- Zones
- RSS Feeds
- Events
- Print Subscriptions

Oracle upgrades Data Warehouse Builder

Paul Roberts, IDG News Service Today's Top Stories > or Other Databases Stories >

March 13, 2003 (IDG News Service) — A new version of Oracle Corp.'s Data Warehouse Builder will make it easier for database developers and administrators to tie together information from different data stores within an enterprise, the company announced yesterday.

Version 9.0.4 of the Oracle9i Data Warehouse Builder (OWB) includes a number of enhancements from previous versions, according to Oracle.

Those enhancements include:

A new SAP Integrator that makes it easier to extract information from SAP R/3 systems by SAP AG.

Better workflow features and a more user-friendly interface that simplifies the process for creating extract, transform and load (ETL) processes.

Support for industry standards such as XML Metadata Interchange and Process Description Language to simplify the process of integrating OWB with third-party products.

Oracle9i Warehouse Builder is used to pull data from disparate sources such as financial and accounting databases into an Oracle9i database. Once data is in a predictable format within Oracle, it can be accessed by more sophisticated analytic tools such as customer relationship management and enterprise resource planning applications, according to Mark Shainman, a senior analyst at Meta Group Inc. OWB competes with similar "native" ETL tools from Microsoft Corp., which offers its Data Transformation Services (DTS), and IBM, which sells Warehouse Manager, according to Shainman.

By making it easier for database developers to pull data into a specific database environment such as Oracle or SQL Server, native ETL tools are competing with more expensive, but "database agnostic" tools such as those by Cognos Inc. and Informatica Corp.

Database vendors are investing heavily in ETL tools, hoping to drive adoption of their databases as platforms for centralized data warehouses, according to Shainman. "People want to know what's going on in their businesses. The database licensing sales and momentum are behind decision support and data warehouses. That's where the spending is," he said.

Gregory Richard, chief technology officer of Canadian company DecisionsToday Inc. in Hull, Quebec, said that his company uses OWB in addition to Microsoft's DTS and Cognos' Decision Stream to build tools that accelerate the process of building data warehouses. Decisions Today's software does analysis of an organization's business databases and systems, automatically generating a data warehouse model and ETL routines to supply it. The software is licensed directly to database vendors and sold to large enterprises, Richard said.

Due to limitations in OWB's scripting environment, Decisions Today was prevented from integrating with prior product editions. Version 9.0.4 of OWB fixed that problem, and Decision Today is now using OWB to build and automate ETL processes for Oracle, Richard said.

The product's user interface is also much improved in the latest version, Richard said. "It used to be quite cumbersome to use. You had to pop in and out of dialog boxes a lot."

The product's new interface brings it more in line with competing products, providing a diagram to represent an overall build and "point and click" support for editing object properties.

Version 9.0.4 runs on a variety of platforms, including Linux, Sun Solaris and Windows. The product is being offered for \$5,000 per Named User Plus as part of the Oracle9i Developer Suite, according to Oracle.

Despite its focus on deploying to the Oracle environment, OWB offers a powerful and relatively inexpensive alternative to other solutions, Richard said. "When you look at price per feature, OWB blows the others away," he said.

Reprinted with permission from



For more news from IDG visit IDG.net
Story copyright 2006 International Data Group. All rights reserved.

TODAY'S TOP STORIES

New Microsoft-compatible VoIP phones to compete with Avaya, Cisco

Microsoft 'desperate,' says patent complaint target OpenOffice.org

Analysis: Microsoft patent claims hint at internal issues

More top stories >

IDG RELATED CONTENT

How to Calculate the Cost of a Data Breach - CIO.com - Business Technology Leadership

CIO.com - Business Technology Leadership - Oracle Plans to Release 37 Patches Next Week

Software manages storage resources - Network World Network World

WEBCASTS

Business Service Management Trends: How a CMDB Drives IT-Business Alignment

Case Study: A manufacturing SMB Gets Improved IT capabilities as well as savings

WHITEPAPERS

Total Cost Comparison: IT Decision-Maker Perspectives on EMC, HP and NetApp Storage Solutions in Enterprise Database Environments

How to Choose the Right Search Engine

Five Differences Your Company Can't Afford to Ignore

COMPUTERWORLD REPORTS

RESOURCE ALERTS

Sign up to receive Security Resource Alerts

SIGN-UP

HANDICAP

Online Office Suites: The Winner Is Clear

In our hands-on review, one free Web-based Office alternative stands head and shoulders above the rest. Read more...

STRATEGY



Strategies for data center automation

Join the Enterprise Strategy Group and Symantec Corporation for an evaluation of available data center automation options and a discussion of the best approach for your company.

Download this on demand webcast now! >

FOCUS

Must-have Firefox extensions

These plug-ins give you souped-up functionality, better look and feel, and streamlined development tasks. And some are just plain cool. Read more...

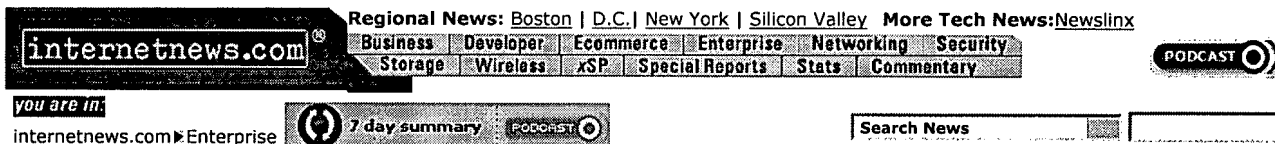
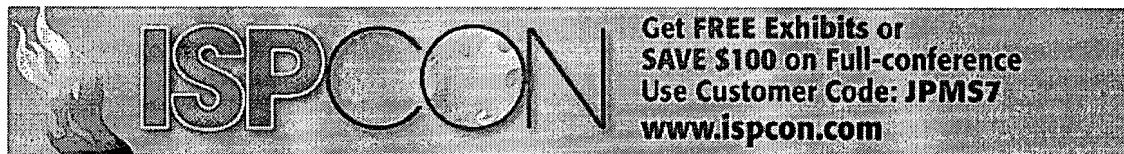


Get your
enterprise
IT news
when you
want it!

Weekly in Print.
Always online.

**SUBSCRIBE
NOW!**





/// BUSINESS.COM

Search

internet.com

Compare prices & save on:

Home Theater

- LCD TVs
- Plasma TVs
- Flat Panel TVs
- Video Projectors
- LCD Projectors
- HDTV Receivers
- Home Audio

Compare Cameras

- Canon Camera
- Kodak Camera
- Nikon Camera
- Fuji Camera
- Sony Camera
- Digital SLR Camera
- Underwater Camera

GPS

- Garmin
- Magellan
- Lowrance
- TomTom
- Navman
- Marine GPS
- Handheld GPS

Mp3 Players

- Ipod Nano
- Ipod Video
- Gigabeat
- Sansa
- Zen Micro
- Mp3 Transmitters
- Ipod Speakers

INTERNET SHOPPER

Compare Prices:

On Demand Webcast: Chilling Changes in the Server Room. The more your organization depends upon the network & your servers, the more heat they produce, the more power they consume, the higher the power costs. Learn more here.

Enterprise

March 12, 2003

Oracle Tweaks Business Intelligence Software

By Clint Boulton

Oracle (Quote) Wednesday revamped a business integration tool that facilitates and manages data in business intelligence applications.

Business intelligence is the gathering of as much pertinent data as possible to make better business choices. Applications that fall under its umbrella include those for decision support systems, query and reporting, online analytical processing (OLAP), statistical analysis, forecasting, and data mining. Redwood Shores, Calif.-based Oracle's competitors in this space include IBM, but other players such as Cognos, Applix and Hyperion provide some aspect of business intelligence software.

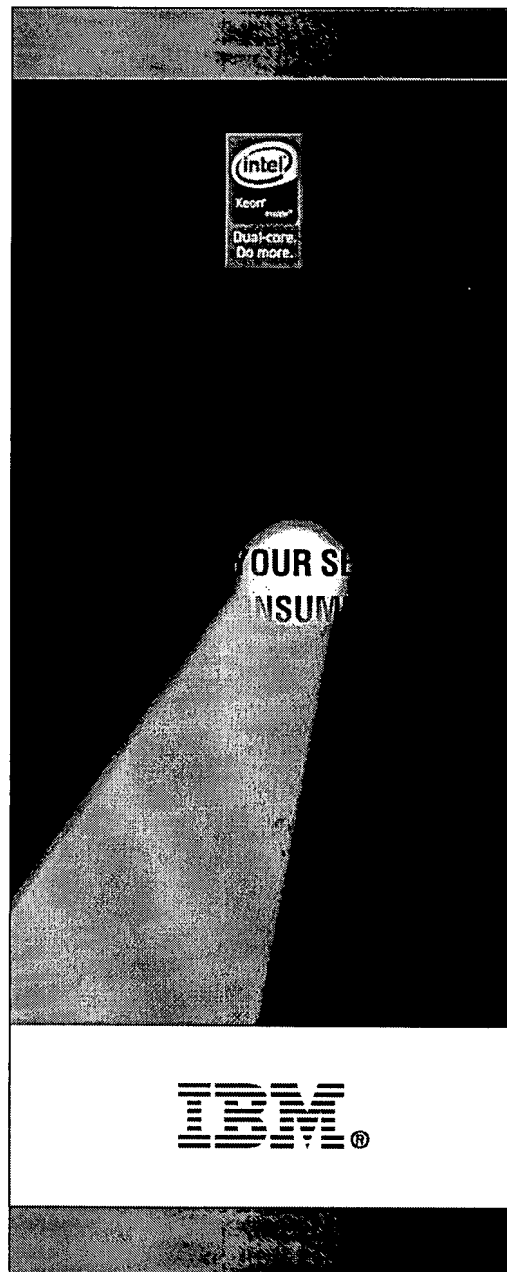
Oracle's new product, Oracle9i Warehouse Builder Version 9.0.4, is an extraction, transformation and loading (ETL) tool that lets database administrators and developers design and build business intelligence applications more efficiently.

Speaking of the integration aspects, Oracle9i Warehouse Builder Version 9.0.4 uses the Oracle Database as its runtime engine, relies on the ETL capabilities and OnLine Analytical Processing (OLAP) engine built into Oracle9i Database Release 2 to create multi-dimensional objects.

As with many enterprise software players, Oracle pledges allegiance to open standards, and in this instance, crafted a product that will work well with competitor's software.

For example, Warehouse Builder Version 9.0.4 supports Microsoft's SQL for data integration routines, Object Management Group's Common Warehouse Meta-Model (OMG CWM) and eXtensible Markup Language Metadata Interchange (XMI) for metadata exchange, as well as the new Processing Description Language (XPDL) for process definition.

In addition to support for mainframes and other



internet.commerce

[Be a Commerce Partner](#)
[Web Hosting](#)
[Computer Deals](#)
[T-Shirts](#)
[Promotional Products](#)
[PDA Phones & Cases](#)
[Compare Prices](#)
[Boat Donations](#)
[Online Degrees](#)
[Help Desks](#)
[Memory Upgrades](#)
[Graphics Cards](#)
[Cheap Plasma TVs](#)
[Marketing Products](#)
[Desktop Computers](#)

Newsletter Signup

☒ Internet Daily

☐ InternetNews Business Report

☐ Boston News

☐ DC News

☐ NY News

☐ SiliconValley News

email

select a newsletter above, type your email and click the arrow to sign up!

internet.com

[Developer](#)
[International](#)
[Internet Lists](#)
[Internet News](#)
[Internet Resources](#)
[IT](#)
[Linux/Open Source](#)
[Personal Technology](#)
[Small Business](#)
[Windows Technology](#)
[xSP Resources](#)
[Search internet.com](#)
[Advertise](#)
[Corporate Info](#)
[Newsletters](#)
[Tech Jobs](#)
[E-mail Offers](#)

databases such as IBM and Sybase, the new tool includes a SAP Integrator that generates code to extract data from SAP R/3 systems and populate data warehouses.

The software also features a Runtime Audit Browser to verify the results of all running jobs and reveal any errors that may have occurred while loading data and a Process Flow Editor that allows users to create ETL transformations and automatically generate code to integrate these and other business processes.

Oracle9i Warehouse Builder Version 9.0.4 is available now for free download and evaluation from Oracle Technology Network. It also can be purchased for \$5,000 per Named User Plus from the Oracle Store as part of Oracle9i Developer Suite.

Tools: [Email](#) [Print](#) [Digg This Story](#)

[del.icio.us](#)

[Enterprise Archives](#) | [7 day summary](#)

Add internetnews.com to your favorites

[MY YAHOO!](#) [Windows Favorites](#)

Add internetnews.com to your browser search box

[IE 7](#) | [Firefox 2.0](#) | [Firefox 1.5.x](#)

Receive news via our XML/RSS feed [XML](#) [RSS](#)

Xeon Processors Packed with Latest Server Technologies

By adding more servers, the datacenter can quickly grow out of control, and bring power and cooling costs with it. Using new server technology available from Intel, however, IT managers can keep costs in check while running a more efficient operation. [Click here.](#)

Applying RFID Technology in High Volume Manufacturing

To explore the impact of radio frequency identification (RFID) technology in manufacturing, Intel's Technology and Manufacturing Group deployed an RFID pilot within Intel's largest semiconductor assembly and test facility. The pilot tracked 80,000 microprocessors from the end of the manufacturing line through Intel's warehouse, into a major customer's warehouse, and onto the customer's factory floor. [Click here.](#)

Building a Real-World Model to Assess Virtualization Platforms

To drive business decisions in the data center, Intel IT created a performance-driven methodology to compare the operating costs of virtualization platforms. By measuring workload performance, platform performance, and power consumption they were able to approximate the total cost of ownership (TCO) of each platform. [Click here.](#)

Building an Enterprise Data Warehouse and Business Intelligence Solution

To achieve end-to-end visibility into critical business functions across the company, Intel IT deployed an integrated warehouse solution. The solution employs a consolidated enterprise data warehouse along with business intelligence applications. Today's EDW occupies over 20TB of usable data space, processes 90 billion rows of data per month, and is accessed by over 11,000 users worldwide. [Click here.](#)

Which Is Right for You Itanium 2 or x86 Architecture?

With the growing market success of Itanium 2-based solutions and the integration of 64-bit extensions into x86-based processors, organizations now have a choice of two widely supported, industry-standard, 64-bit architectures. Which is the right choice for your data center? [Click here.](#)

[Visit the Intel Resource Center](#)

Special Reports

Technology in the Courts



From patent infringement and company theft to fraud and litigation suits brought on all levels of the courts, the tech industry gives the law a spin. [\[more \]](#)

Stats

- [Report: Oracle Commands Database Software](#)
- [Apple's Green Road Paved With iPhone Takers](#)
- [Broadcasters Hot For Online Classifieds](#)
- [AMD's Slip Gives Intel Some Running Room](#)
- [Teens Are Protecting Themselves Online](#)
- [U.S Slips To Third In Worldwide PC Market](#)
- [Banks Shell Out The Bucks For Compliance](#)
- [U.S. Economy to IT: Thank You](#)
- [Storage Sales Slow](#)
- [Hitting The March Madness Bandwidth Hoop](#)
- [Online Ad Dollars Stay Hot](#)
- [Bet You Don't Know Who I Really Am](#)

Hot Topics

- [GPL: Software Freedom Redefined](#)
- [Oracle's Linux Realm](#)
- [Breaching Your Data](#)
- [SaaS in The Market](#)
- [Stock Option Backdating Scandals](#)
- [The SMB Holy Grail](#)

Top Stories

- [AGs Ask MySpace For Predator Data](#)
- [HDS Embraces SOA With New Storage Array](#)
- [Microsoft: Open Source Infringes on 235 Patents](#)

Recent Headlines

Enterprise News

- [HDS Embraces SOA With New Storage Array](#)
- [OpenSEA Adds Open Source Spice to 802.1x](#)
- [CodeGear Gunning For Enterprise Ruby on Rails](#)

[Skip Headers](#)

Oracle9i Warehouse Builder User's Guide
Release 2 (9.0.4)
Part Number B10657-01

Go To Table	Go To
Of Contents	Index
Contents	Index

[Go to previous page](#) [Go to next page](#)

14

Importing and Exporting with the Metadata Loader (MDL)

- 2) The Metadata Loader (MDL) enables you to populate a new repository as well as transfer, update, or restore a backup of existing repository metadata. You can also take snapshots of your metadata and use them for backup, compare, and restore purposes.

This section contains the following topics:

- [Overview of Import and Export Using Metadata Loader](#)
- [Importing and Exporting Metadata Using the Metadata Loader](#)
- [Exporting Metadata](#)
- [Importing Metadata](#)
- [Using the Metadata Loader Command Line Utility](#)
- [Splitter for Exporting and Importing Warehouse Builder Mappings](#)

Overview of Import and Export Using Metadata Loader

Warehouse Builder provides several features that enable you to copy and move metadata for the purposes of backup, history management and version management.

- 2) You can import and export metadata for any type of object on the navigation tree using the Metadata Loader (MDL) utility. Access MDL through the Warehouse Builder client or through the OMB Plus scripting interface. Use the import and export functionality to backup metadata or to migrate metadata when upgrading Warehouse Builder.

You can also then move exported files into a third-party version control tool such as Oracle Repository, ClearCase, or SourceSafe. If you enter version numbers in your project properties, it is easier to track your export and import versions in this setting.

You can also perform metadata change management by taking snapshots of your metadata using the

OMB Plus scripting interface. Snapshots enable you to capture definitions of metadata objects using Warehouse Builder scripts. Use snapshots for the purpose of metadata backup and version management. For more information about metadata change management, see [Chapter 15, "Metadata Change Management"](#).

Importing and Exporting Metadata Using the Metadata Loader

The MDL enables you to copy or move metadata objects between repositories, even if those repositories reside on platforms with different operating systems.

The MDL consists of two utilities: metadata export and metadata import. The export utility extracts metadata objects from a repository and writes the information into a text file. The import utility reads the metadata information from an exported text file and inserts the metadata objects into a repository. MDL uses its own format, and the MDL import utility only reads files of MDL format (files created by MDL Export or MDL File Upgrade).

You can operate the MDL from the Warehouse Builder console or by using a command line interface. For instructions on using MDL through the command line interface, refer to ["Using the Metadata Loader Command Line Utility"](#). If you use the console menu, a graphical interface guides you through the export or import processes.

Use the metadata loader to perform any of the following tasks:

- **Backup metadata:** The MDL is an important part of your disaster recovery strategy. You can export a file with your existing repository metadata as a backup and use that exported file to restore a repository if necessary.
- **Seed a new repository:** You can export data from an existing repository and use it as the basis for a new repository.
- **Migrate a repository:** You can export metadata to a file and then re-import it. This is commonly done when upgrading to a newer version of Warehouse Builder. For details on upgrading from a previous version of Warehouse Builder, see the *Oracle9i Warehouse Builder Installation and Configuration Guide*.
- **Copy metadata:** A multiple user development environment can result in multiple copies of the same metadata. The MDL makes it easy to load single set metadata into more than one repository.

This section contains the following topics:

- [Required Access Privileges for MDL](#)
- [About Metadata Loader Results](#)
- [About the Metadata Loader Log File](#)

Required Access Privileges for MDL

The Warehouse Builder repository allows multiple clients to access the same repository schema concurrently. Warehouse Builder uses locks to allow only one client to access to change repository objects. While an object is locked, other clients can only view it as it existed after the last transaction instigated by any user is committed.

Tip:

To ensure that you are exporting the most up-to-date metadata, you need to be the sole client accessing the repository.

If you click **OK** when prompted, the MDL commits changes made to the repository after a successful metadata import (any import with no error messages, including imports with only information or warning messages). The MDL also executes a rollback after an unsuccessful import. This means that Warehouse Builder attempts to acquire one lock for each primary object (an object in the first level on the navigation tree) in the repository that matches an object in the MDL file. These objects include, but are not limited to, projects, modules, and tables--individual columns are not locked. Therefore, you must be able to hold the locks for these objects while you import metadata. If other users hold locks for objects to which you are importing, the MDL will fail.

Tip:

To ensure a successful metadata import, you need to be the sole client accessing the repository.

If the MDL import affects too many objects in the repository, the MDL automatically switches to single user mode. This means that no other users can log on to the repository until after the MDL import completes. Single-user mode allows the MDL to avoid the performance degradation that results from using a large number of locks. In single-user mode, the MDL is less likely to deplete the repository enqueue resources. If other users are logged into this repository when MDL attempts to switch to single-user mode, MDL cannot switch to single-user mode and subsequently fails.

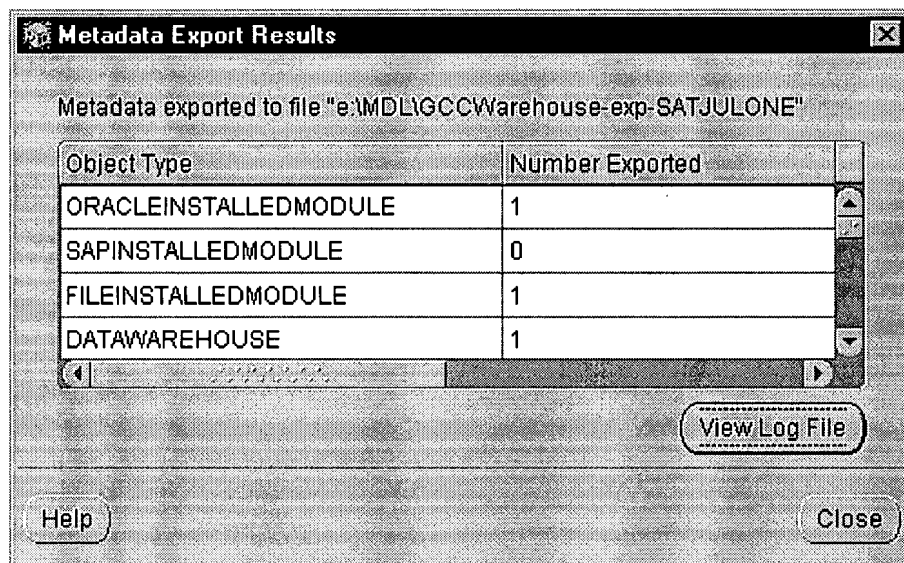
You also need to have MDL_IMPORT security privileges in order to import metadata. For more information on security, see ["Managing Security with PL/SQL"](#).

About Metadata Loader Results

Each time you use the export or import utilities, the MDL reports the results of an action and writes diagnostic and statistical information to a log file.

The MDL reports the results after any import or export task with a dialog. If you want detailed information, you can view a detailed log by clicking **View Log File** from the Metadata Export Results dialog as shown in [Figure 14-1](#).

Figure 14-1 Metadata Export Results



Text description of the illustration meuexpre.gif

The results dialog lists the metadata objects found in either the file or the repository, and the number of each object that was exported or imported. You can use the results dialog to ensure that all of the objects were exported or imported. The MDL identifies the objects that were exported or imported and compares it with the eligible objects list. A zero in the Number Exported or Number Imported column for any object indicates that the MDL found no object of that type in the repository. However, if a zero appears for any object that exists in the repository or imported MDL file, then MDL encountered a problem when importing or exporting that object.

About the Metadata Loader Log File

Whenever you export or import repository metadata, the MDL writes diagnostic and statistical information to a log file. By default, the log file is located in the directory and path specified in the Message Log tab located in the Preferences dialog. You can specify an alternative location for the log file when invoking MDL.

Example 14-1 displays the contents of a typical import log file.

Example 14-1 Log File Showing Import Results

```
Import started at 04/25/2001 4:59:46 PM
*****
* Import for OWB Release: 3.0.0.0.0 Version: 3.0.0.3.0
* User: user30_3i Connect String: epaglina-pc:1521:ora8i
* Data File: d:\owb3000\sco_dim_time_phy_m_tgt.mdl
* Log File: d:\owb3000\imp_dim_time_phy_m_tgt.log
* Trace: B
* Trace File: d:\owb3000\imp_dim_time_phy_m_tgt.trc
* Physical Names: Y Mode: CREATE
* Ignore Universal Identifier: Y Commit At End: Y
*****
Informational at line 15: MDL-1207 PROJECT with physical name <PRJ_Dimension> not impor
because it already exists.
Informational at line 21: MDL-1207 DATAWAREHOUSE with physical name <WH> not imported
because it already exists.
Informational: MDL-1134 COMMIT issued at end of import data file.
```



```
Counts for OWB Import Utility
-----

Total Projects Processed by Import = 1

-----

Project = PRJ_Dimension

Entity in Project          Added      Replaced      Skipped
-----
DATAWAREHOUSE:             0           0           1
DIMENSION:                 1           0           0
LEVEL:                     3           0           0
HIERARCHY:                 2           0           0
LEVELRELATIONSHIP:         4           0           0
COLUMN:                   18           0           0
UNIQUEKEY:                 8           0           0
PRIMARYKEY:                2           0           0
CONFIGPARAM:              60           0          60
CHILDCONFIG:              14           0           0

Import ended at 04/25/2001 4:59:52 PM
```

The log file enables you to monitor and troubleshoot export and import activities in detail. The log file contains the following types of status messages:

- **Informational:** Provides information about the import or export, such as missing metadata objects, whether or not objects were imported, and any reasons why they were not imported or exported.
- **Warning:** Cautions you about the import or export of an object but does not indicate a failed or aborted load. A warning notifies you of the possibility of unexpected load results for the load.
- **Error:** Indicates that the MDL export or import was aborted and did not complete successfully. The error message offers a brief reason for the failure.

This log also displays the total number of objects that have been added, replaced, and skipped. A zero in any column for any object indicates that the MDL found no object of that type in the repository or in the imported MDL file.

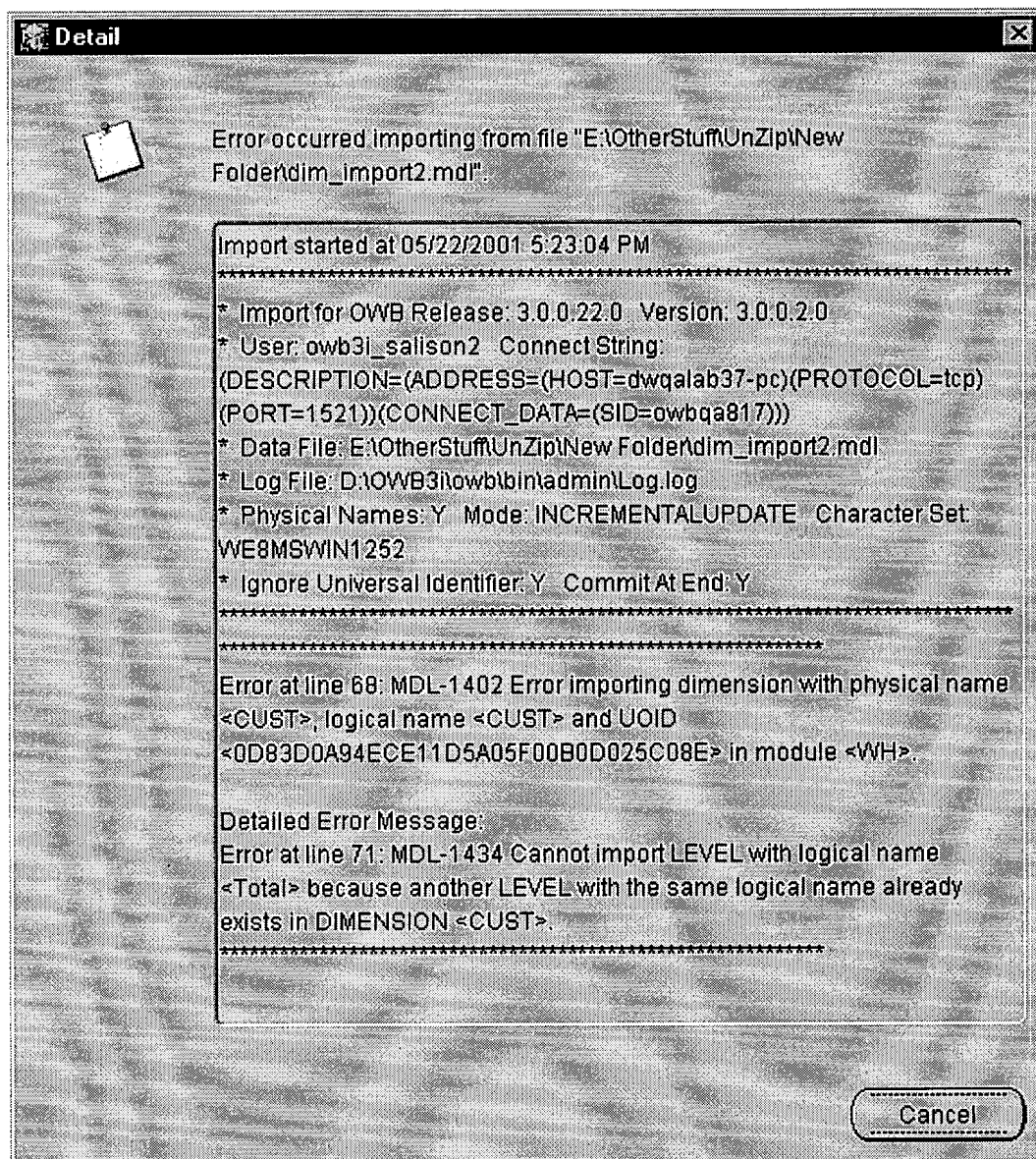
Detailed Error Logs

If you are running an MDL Import and encounter an error, an error message displays.

Click **Detail** to display a detailed error log that lists the repository object and the object line in which the error occurred. Detailed messages are useful whenever you import metadata into repositories with existing metadata because they alert you to problems such as improperly defined metadata objects and object duplication.

Figure 14-2 is an example of a detailed error message.

Figure 14-2 Detailed Error Message



Text description of the illustration det_err.gif

In this example, the repository object is the CUST dimension and the imported object is the TOTAL level. The dialog explains that the TOTAL level cannot be imported into the CUST dimension because a level named TOTAL already exists.

Exporting Metadata

The Metadata Loader can export all repository objects. The MDL also exports information belonging to metadata objects such as table columns and their constraints, data loading configuration parameters, and named attribute sets. You can use the MDL to export an entire project or a subset of objects within a project.

When you export repository metadata, the Metadata Loader writes the extracted metadata to a delimited text file. The MDL stores this file outside the repository by assigning a default path and file name to the exported MDL file.

This section contains the following topics:

- [Before Exporting Metadata](#)
- [About the Metadata Export Utility](#)
- [Exporting Metadata using Warehouse Builder Client](#)
- [Metadata Export File Format](#)
- [Archiving a Project](#)

Before Exporting Metadata

Before you attempt to export metadata, ensure you have the following:

- **Required access privileges.** To ensure that you are exporting the most up-to-date metadata, verify that you are the sole client accessing the repository in read/write mode. For more details, see ["Required Access Privileges for MDL"](#).
- **Sufficient disk storage.** If you lack sufficient disk space on the machine to which you export the metadata, the export fails. Your destination machine must be able to contain the entire metadata file. The export utility cannot save portions of the metadata file.

About the Metadata Export Utility

You can export metadata from a Warehouse Builder repository using one of the following:

- **Metadata Loader command line utility.** You can use the command line utility to perform tasks additional tasks not available from the client interface. For instructions on exporting from the command line, see ["Using the Metadata Loader Command Line Utility"](#).
- **Warehouse Builder client interface.** For instructions on using the client interface, see ["Exporting Metadata"](#).

Using the command line or the client interface, you can export an entire project, collection, or module, or any subset of objects. If you export a subset of objects, the MDL exports definitions for each object you have selected and the parent objects to which the subset belongs. This enables the MDL to maintain the tree relationships for those objects during metadata import.

For example, if you export a single dimension, the export file contains definitions for:

- The dimension (and its hierarchies and levels)
- The dimension node to which the dimension belongs
- The module to which the dimension node belongs
- The project to which the module belongs

If you are exporting a subset of objects, make sure you export all referenced objects and import them as

well. The Metadata Import Utility allows you to import repository objects even if the references for those objects cannot be satisfied.

For example, if you export a cube, the foreign key references will be exported, but the dimensions to which they refer will not. If the metadata in the dimension tables changed, then the foreign key references imported to the new repository would become incorrect.

Exporting Metadata using Warehouse Builder Client

Use the metadata export utility to export objects from a Warehouse Builder repository into an MDL file.

To export metadata from a repository using the Warehouse Builder client interface:

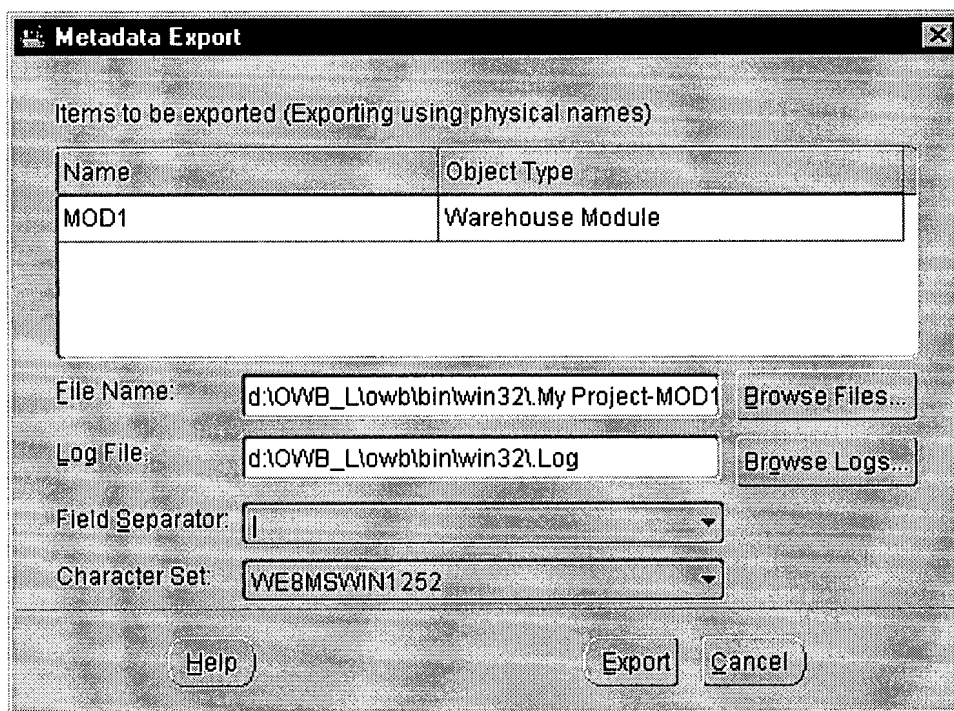
1. From the Warehouse Builder Console, select the object or objects you want to export.

You can export individual objects such as tables or groups of objects. When you export projects nodes, or modules, you also export the objects they contain. When you export collections, you also export the objects they reference.

2. From the **Project** menu, select **Metadata Export** and then **File**.

The Metadata Export dialog displays the names and types of the objects you are exporting. The Metadata Export dialog also displays default settings for the export file as shown in [Figure 14-3](#).

Figure 14-3 Metadata Export Dialog



Text description of the illustration meuexpmo.gif

3. You can accept or change the following default settings:

- **File Name:** Type the name of the export file to create or click **Browse** to locate a directory or file. The filename you assign must end with .mdl.
- **Log File:** Warehouse Builder records information about the export in a log file. You can change this location by entering a new path and filename. Type the path and file name in the field or click **Browse** to locate a directory or filename.
- **Field Separator:** Table fields in the export file are separated with a pipe (|) by default. If your file already has the pipe (|) symbol as part of its data, you can change the default field separator to a caret (^) by selecting it from the list.
- **Character Set:** Select the character set to use in the export file. The default character set is defined by the Warehouse Builder client system. Use the list to change the output character set.

4. Click **Export**.

If you made changes to the repository metadata prior to running the export utility, the Metadata Export Confirmation dialog displays. Click **Commit** to save changes or **Rollback** to revert to the previously saved version. You must have read-write access to the repository in which you are exporting metadata to commit changes.

The Metadata Export Progress dialog displays the progress. When the export completes, the Metadata Export Results dialog displays.

Click **View Log File** for a detailed view of the export process.

Metadata Export File Format

The Metadata Loader formats the .mdl export file using keywords and position as shown in Example 14-2.

Example 14-2 Sample Records from an Export File

```
#Project data <PhysicalName> <LogicalName> <UniversalID> <Version Label>
PROJECT|WarehouseName|Warehouse Name|A86184D5336911D58E9000B0D02A59E4|null
#Dimension <PhysicalName> <LogicalName> <UniversalID> <Prefix> <UsageType> <Import
<Generated>
DIMENSION|Channels|Channels Dimension Data
Mart|7E727655029911D58DC900C04F48E9ED|ch|null|N|N
```

In this example, each record in the file begins with a keyword followed by one or more variable-length fields. Table fields are separated by a pipe (|) by default.

Archiving a Project

Archiving a project allows you to copy metadata stored within a Warehouse Builder repository to an external location for the purpose of securing that data at a fixed point in time. Warehouse Builder provides an Archive Wizard to assist you in this process. The Archive and Restore utilities initially write to a file system. You can then move files from this file system into a third-party

version control tool such as Oracle Repository, ClearCase, or SourceSafe.

Note:

The Archive and Restore utilities will be desupported in the next release of Oracle9i Warehouse Builder.

You must set up your Archive/Restore settings on the Preferences page before you can archive or restore your project. If you attempt to archive or restore without setting these preferences, you get an error.

Project Version Labels

Before you archive your project, you can update the project version label with the Project Properties dialog. There are two places in Warehouse Builder where you can set up the version label used in the archive/restore:

- The first is in the New Project Wizard. The New Project Wizard contains a step that allows you to define version properties. The version label that you set here is the version label that is used when that project is archived.
- After you have created a project, you can edit the version label by opening the Properties dialog for the project. Click the Version Properties tab to modify the project version label.

Differences Between Archive and Export

Archive and Restore are different from Import and Export. [Table 14-1](#) describes the differences between Archive and Export.

Table 14-1 Differences Between Archive and Export

Feature	Archive	Export
Character Set	UTF8	User Configured
Field Separator	Pipe Character ()	User Configured
Read-only Detection	Detects and prompts you to re-try	Detects and then fails
Dump Format	MDL	MDL

Feature	Archive	Export
Log File Name	Generated	Generated and User configured
File Location	Configured by preferences in the following structure: \$ARCHIVE_HOME/ project_name/Label/ Archive_Name	User-defined

Archiving a Project

Before you archive your project, you can update the project version label with the Project Properties dialog (see "[Project Version Labels](#)").

To archive a project:

1. Select Archive from the Project menu.

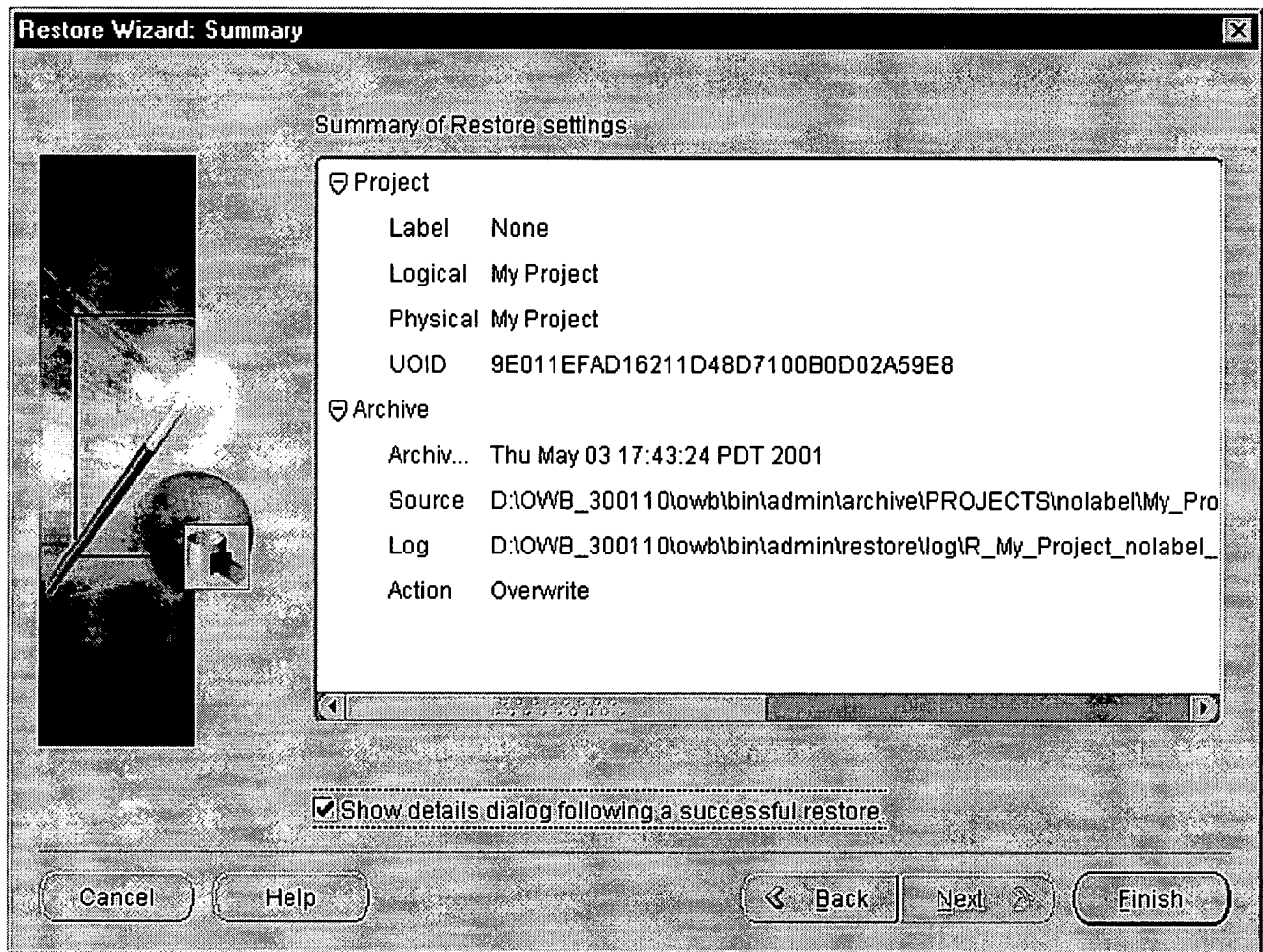
You can also select Archive from the right-click menu when a project is selected.

The Archive Wizard Welcome page displays.

2. Click **Next**.

The Summary page displays a summary of the archive settings prior to running the archive process. If you want to see the details of your archive after the archive process is complete, check the Show details dialog following a successful archive box.

Figure 14-4 Archive Wizard Summary Page



Text description of the illustration rest_wiz.gif

Note:

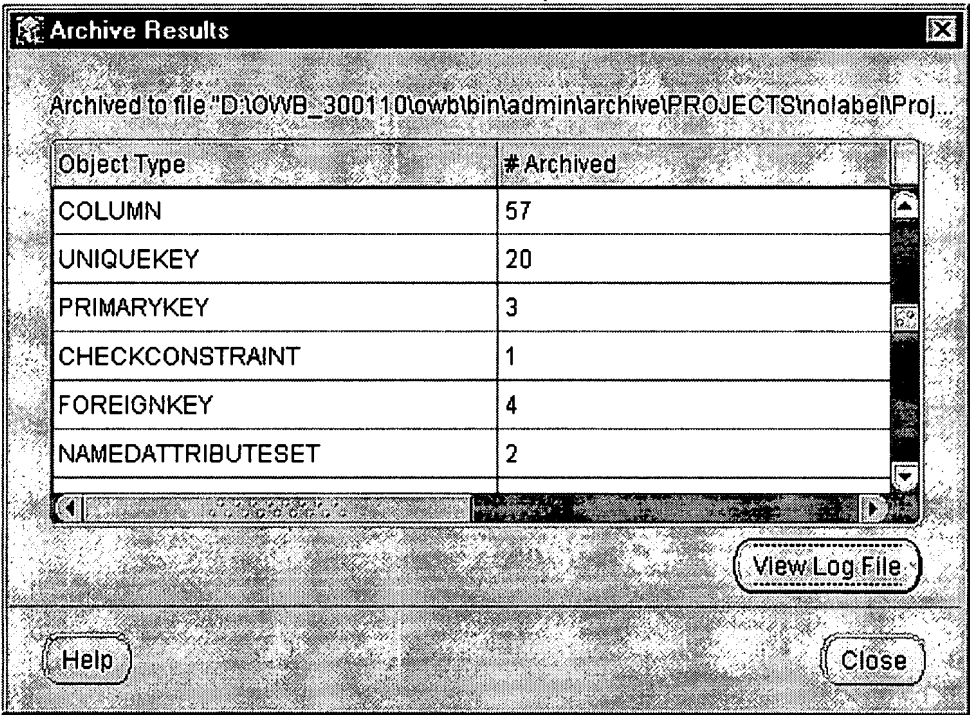
No changes can be made from the wizard. If you notice an error in the Archive Wizard Summary page, click **Cancel** and make the appropriate changes to your Archive/Restore Preferences before continuing with the archive.

3. Click **Finish**.

This begins the archive process. A progress window appears. When the progress bar reaches 100%, the archive process is complete.

If you checked the Show details dialog following a successful archive box, the Archive Results dialog displays. This dialog displays the name of each object type and how many of each were archived.

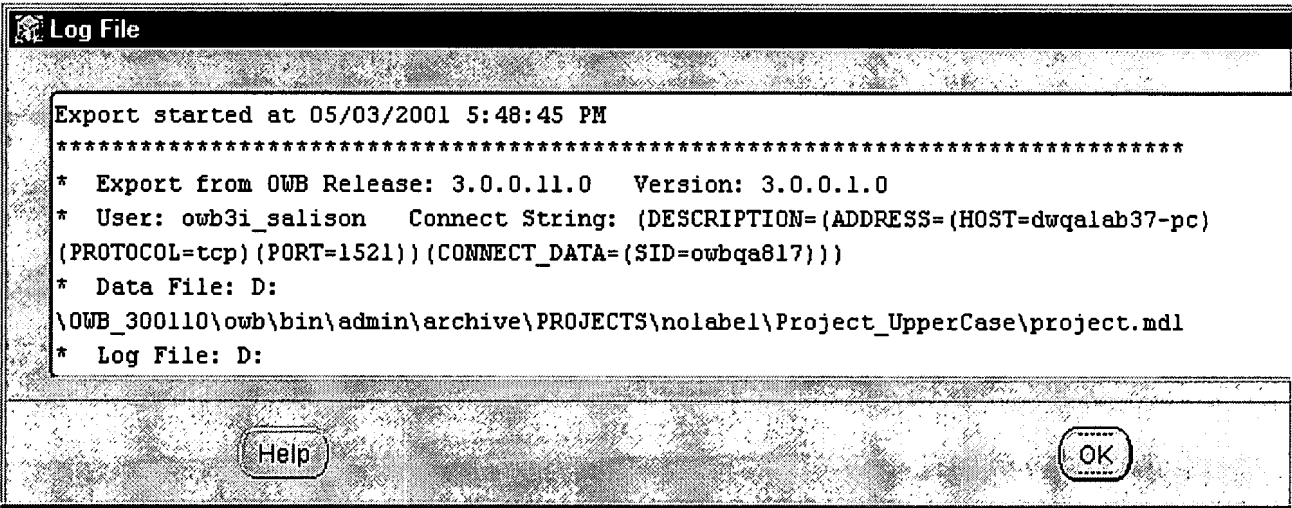
Figure 14-5 Archive Results



Text description of the illustration arch_rsl.gif

For a more detailed look at the archive process, click **View Log File**. This displays the entire log file.

Figure 14-6 Archive Log File



Text description of the illustration arch_log.gif

Importing Metadata

The import utility reads the metadata information from an exported text file and inserts the metadata objects into a repository. The metadata import utility only reads files created by the metadata export utility.

The MDL imports information belonging to exported metadata objects such as table columns and their constraints, data loading configuration parameters, and named attribute sets. You can use the MDL to import objects into a project or a collection.

The following sections describe how to use the Metadata Import Utility:

- [Before Importing Metadata](#)
- [About the Metadata Import Utility](#)
- [Validation Rules Governing Import](#)
- [Importing Metadata using Warehouse Builder Client](#)
- [Restoring a Project](#)

Before Importing Metadata

Before you attempt to import metadata, ensure you have the following:

- **Required security privileges:** You need the MDL_IMPORT privilege before you begin an import. For more information on security, see ["Managing Security with PL/SQL"](#).
- **Required access privileges:** Only a user with read/write access can use the metadata loader import utility. Because the import utility is altering the repository, the metadata objects must be locked prior to importing. For more details, see ["Required Access Privileges for MDL"](#).
- **A backup of your current repository:** Consider taking a backup of your existing repository (either in the form of an export or a metadata snapshot) before attempting a large or complex import. For more information on exporting metadata, see ["Exporting Metadata"](#). For more information on metadata snapshots, see [Chapter 15, "Metadata Change Management"](#).
- **Multiple Language Support base language compatibility:** The base language is the default language used in the repository and is set using the Repository Assistant during installation. This setting cannot be altered after installing the repository. Loading differing base language metadata objects into a repository results in error. For more information on setting the base language in a repository, see the *Oracle9i Warehouse Builder Installation and Configuration Guide*.

About the Metadata Import Utility

You can import metadata into a Warehouse Builder repository using one of the following:

- **Metadata Loader Command Line Utility:** You can use the command line utility to perform tasks additional tasks not available from the client interface. For example, you can override default values for configuration parameters for loading data. For instructions on importing from the command line, see ["Using the Metadata Loader Command Line Utility"](#).
- **Warehouse Builder Client Interface:** For instructions on using the client interface, see

"Importing Metadata".

Validation Rules Governing Import

When you import a set of definitions from exported metadata, the import utility can update existing definitions in a Warehouse Builder project. However, certain metadata definitions require attention to ensure that they are updated. The following are examples of some of the errors you can see:

- **Mapping Definitions.** The Metadata Import Utility does not bind imported mapping operators to their physical objects. If a mapping definition appearing in the source MDL file replaces a mapping definition in the target repository, then the new repository mapping definition is unbound. You may need to reconcile new mapping operators with the physical objects they represent. The MDL generates a warning message to the log file stating that the mapping operators are not bound.
- **Foreign Key Definitions.** It is possible that a source MDL file can contain foreign key references to unique or primary keys that are not in the target repository. If the referenced unique or primary keys for any foreign key appearing in the MDL file does not exist in the target repository, the MDL will generate a warning message in the log file. This message will state that the repository does not contain a referenced key for the foreign key.

Importing Metadata using Warehouse Builder Client

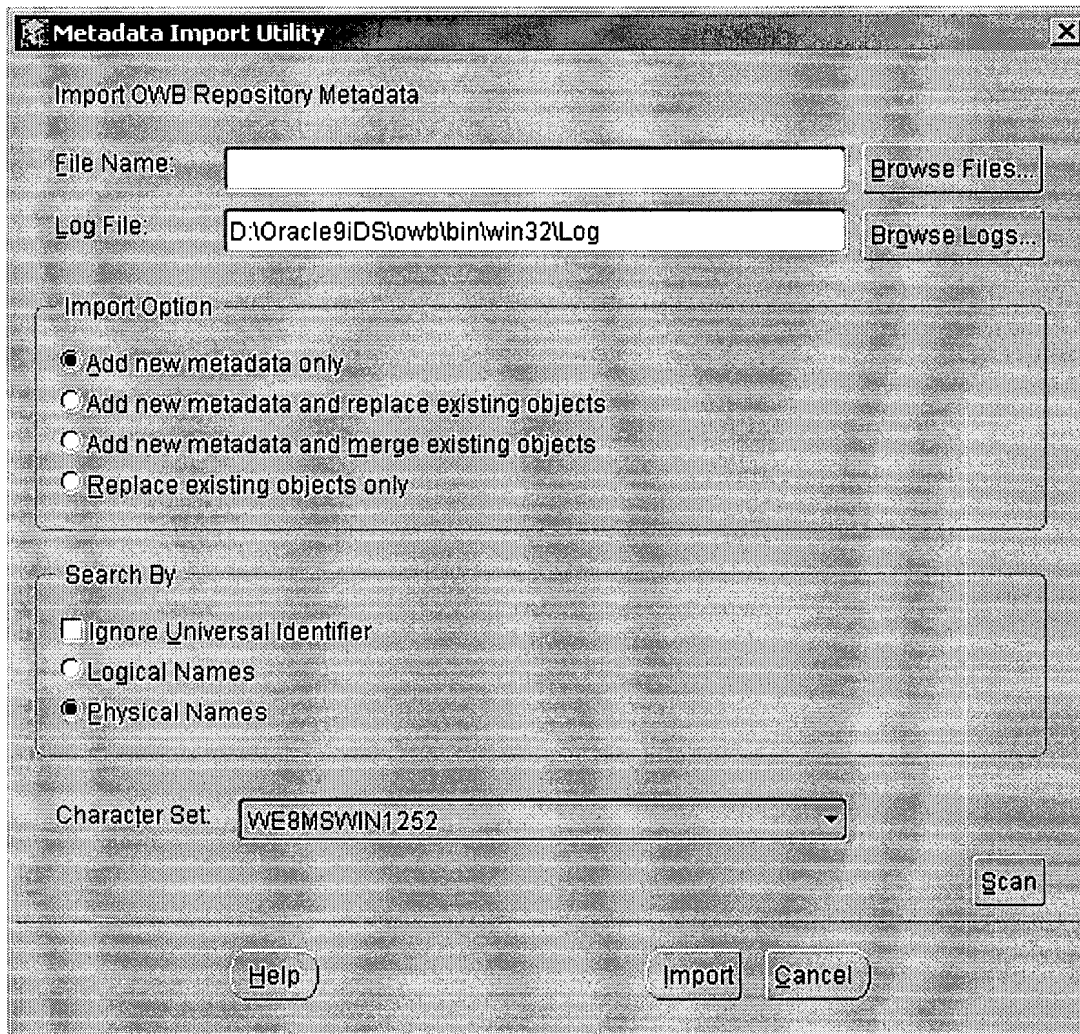
Use the metadata import utility to import objects from an MDL file into a Warehouse Builder repository.

To import objects from an export file using the Warehouse Builder client:

1. Select the project to which you will import metadata.
2. From the Warehouse Builder Console, select **Project** and select **MetaData Import**.

Warehouse Builder displays the Metadata Utility Import dialog as shown in [Figure 14-7](#).

Figure 14-7 Metadata Import Utility



Text description of the illustration miupsibn.gif

3. Specify the names and locations for the import file and its log:

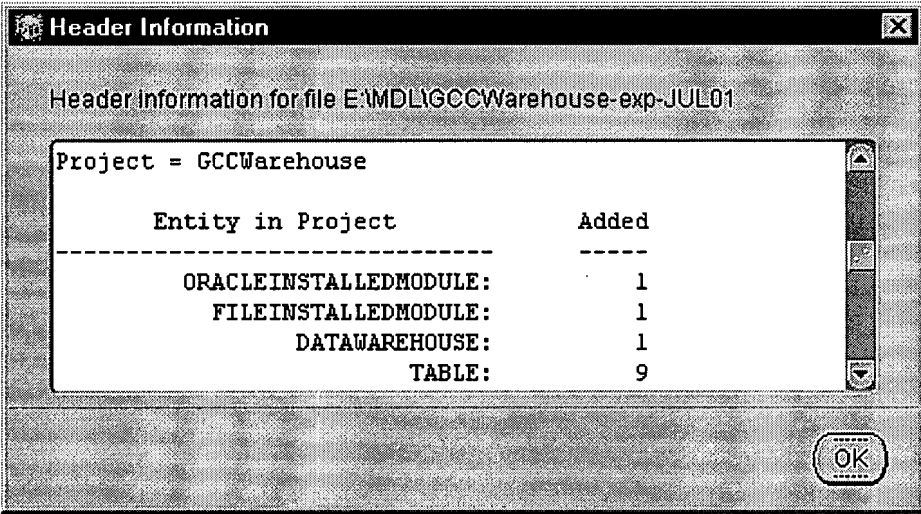
- **File Name:** Type the name of the MDL file or click **Browse** to find the MDL file you want to import.
- **Log File:** Warehouse Builder records information about the import in a log file. You can change this location by entering a new path and filename. Type the path and file name in the field or click **Browse** to locate a directory or filename.

4. Select an **Import Option**. For more information on import options, see "Import Modes". You can select from the following import options:

- **Add new metadata only:** Adds new objects to a repository.
- **Add new metadata and replace existing objects:** Adds new objects to a repository and replaces existing objects.
- **Add new metadata and merge existing objects:** Adds new objects and merges columns into existing objects in your repository.

- **Replace existing objects only:** Replaces existing objects in your repository.
5. In **Match By**, specify the matching criteria the utility uses to compare the metadata in the import file against the metadata existing repository. For more information, see "[Metadata Matching Criteria](#)".
- **Ignore Universal Identifier:** The import utility does not use Universal Identifiers to search for objects you are importing.
 - **Name:** Searches your repository using the physical names of the objects you are importing to make sure the objects do not already exist.
 - **Character Set:** Select the type of character set used to create the import file. The default character set is defined by the Warehouse Builder client machine. Use the drop-down list to change the output character set.
- You can add new languages and character sets using the Repository Assistant. For more information, refer to the *Oracle9i Warehouse Builder Installation and Configuration Guide*.
6. Click **Scan** to display the exported metadata header information as shown in [Figure 14-8](#). The Header Information dialog displays a summary of the total number of object types contained in the metadata file you selected.

Figure 14-8 Header Information

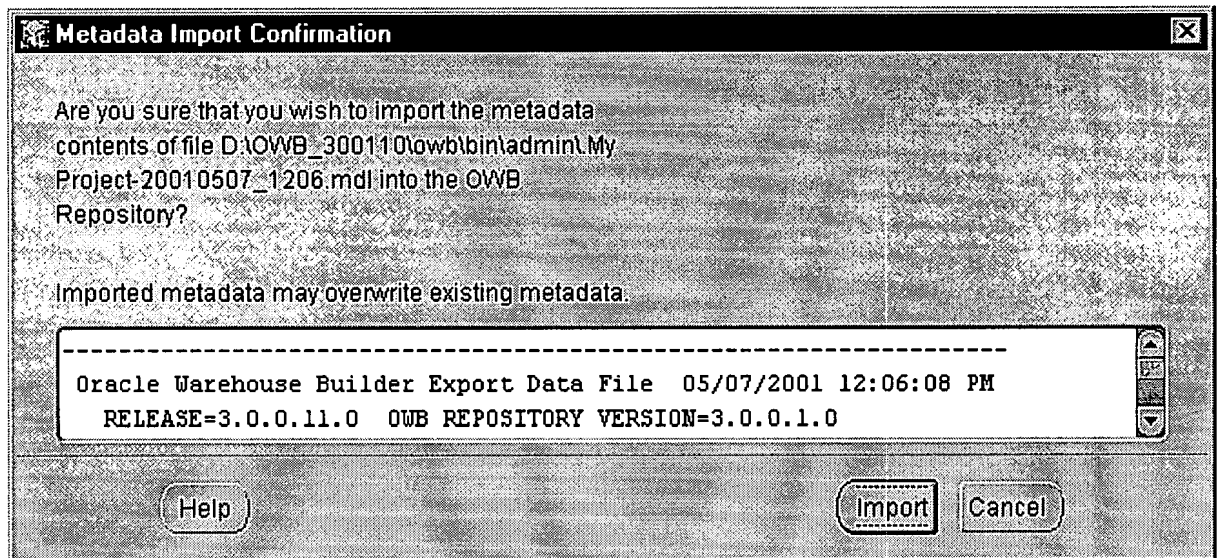


Text description of the illustration miuimpssc.gif

7. Click **Import**.
- If you have made any changes before starting the import, the Metadata Import Confirmation dialog displays. Click Commit to save any changes or Rollback to ignore changes and revert to the previously saved version.
- The Metadata Import Confirmation dialog displays as shown in [Figure 14-9](#) if the

exported metadata data information has not been reviewed.

Figure 14-9 Metadata Import Confirmation

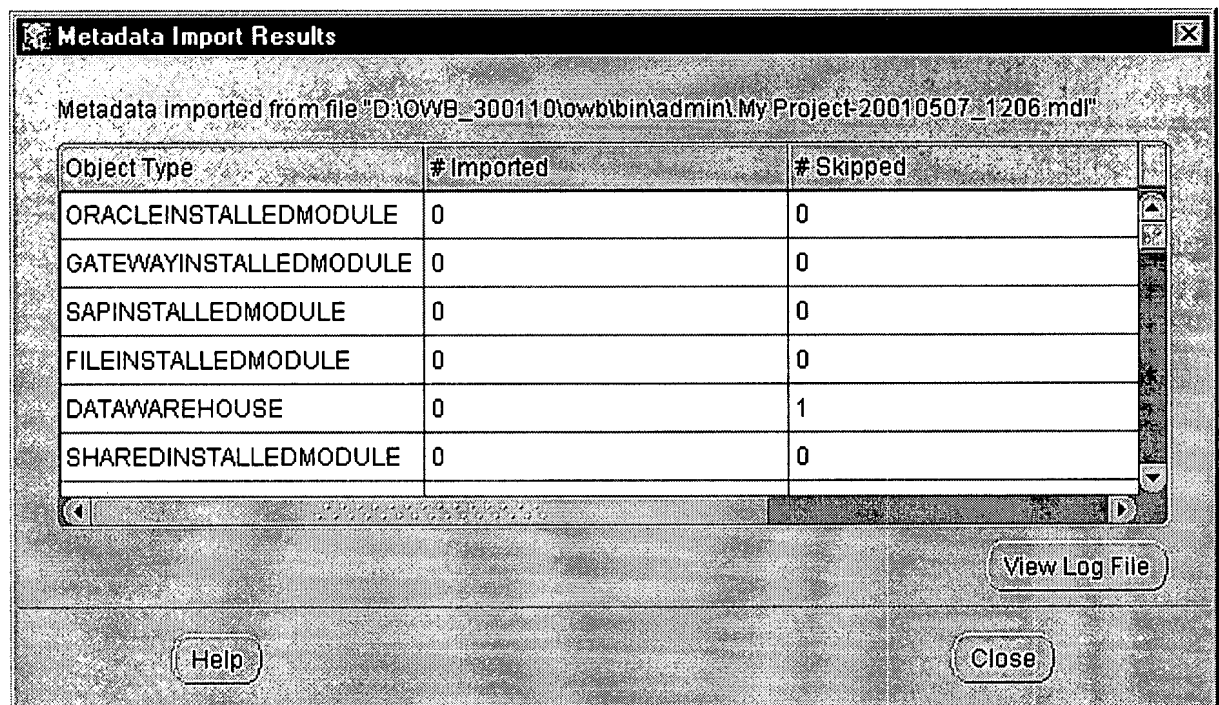


Text description of the illustration mdi_conf.gif

8. Click Import to continue.

The Metadata Import Progress panel displays.

Figure 14-10 Metadata Import Results



Text description of the illustration mdi_rslt.gif

This dialog displays the object types and the number of each type that were imported or skipped. For a detailed view of the import process, click **View Log File**.

Import Modes

The graphical user interface for the metadata import utility operates in one of the following modes:

- 1) • **Add new metadata only (Create Mode):** Adds new objects to a repository. If objects from the MDL file already exist in the repository, they remain unchanged.
- **Add new metadata and replace existing objects (Update Mode):** Adds new objects to a repository and overwrites existing objects with those in the MDL file.
- **Add new metadata and merge existing objects (Merge Mode):** Adds new objects and overwrites existing objects in your repository only if they differ with those in the MDL file.
- **Replace existing objects only (Replace Mode):** Replaces only existing objects in your repository. When importing metadata objects, the MDL will overwrite any existing metadata unless you use this mode.

When you import using the Update or the Replace modes, the import completely replaces the existing object's children so that the final object is exactly the same as the source object. Any existing children of a repository object that are not replaced or added are deleted. This occurs regardless of whether a child object occurs in a mapping or is a foreign, primary, or unique key column in a table or view.

For example, in the MDL export file, the CUST table contains three columns with the physical names: Last_Name, First_Name, and Middle_Init. In the repository, the same table already exists, and contains four columns with the physical names: Last_Name, First_Name, Status, and license_ID. During a replace operation, the columns Last_Name and First_Name are replaced, column Middle_Init are added, and column Status and license_ID are deleted. The final result is that the CUST table in the Warehouse Builder repository contains the same metadata from the CUST table in the export file.

Tip:

Using the replace mode can result in lost data constraints, metadata physical property settings, data loading properties, and mapping definitions. If you choose to use replace mode, ensure that you can restore your repository from backup to its state prior to importing in replace mode.

Metadata Matching Criteria

When you use the metadata import utility, it first searches the repository for metadata

objects that exist in the repository and compares them to those in the file you are importing. How the comparison is made is determined by the loading mode and by the search method you choose. The following methods are available:

- **Physical Name:** Physical names are exported to the export file. The physical name determines whether an object needs to be created, replaced, or merged during an import operation. Use this method when object names in the target directory change, and you want to create new UOIDs for those objects.
- **Universal Object Identifier:** The metadata export utility assigns a unique system-generated identifier to each exported row object called the Universal Object Identifiers or UOIDs. The purpose of the UOID for a row object is to uniquely identify it in an object table. The MDL import utility uses these UOIDs to determine whether a row object needs to be created, replaced, or merged during an import operation. Use this method if you want to maintain UOIDs across different repositories even when object names in the target repository have changed.

By default, the import utility searches by UOIDs. However, the import utility ignores the UOIDs for mappings in the MDL file that already exist in the target repository.

Note:

MDL imports that run in merge mode must use UOIDs for the search criteria in order to merge into existing mappings. Also, if the mapping in the MDL file does not have a Universal Identifier, the mapping cannot be merged into a mapping that matches by name. For more information, see ["Import Modes"](#).

Each search method can be combined with an import mode in several different combinations. Each combination can offer different results in the import process. The mode that you select determines how the metadata import utility will search for metadata objects in the repository prior to importing.

For example, if the search is by the logical name of a repository object in the export file, the Metadata Import Utility searches the repository for the object's logical name. If an object with the corresponding logical name is not found, the resulting actions are based on the import mode you select.

[Table 14-2](#) describes what happens in the available import modes for repository objects that do not match the MDL file names.

Table 14-2 Import Mode without Matching Names

Import Mode	Result
Create	A new object is created.

Import Mode	Result
Mode	
Replace Mode	A warning message is written to the log file that the object cannot be found to replace and the object is skipped.
Update Mode	A new object is created.
Merge Mode	A new object is created.

Table 14-3 describes what happens in the available import modes for repository objects that match the MDL file names.

Table 14-3 Import Mode with Matching Names

Import Mode	Result
Create Mode	A message is written to the log file that the object already exists and the object is skipped.
Replace Mode	The object is replaced.
Update Mode	The object is replaced.
Merge Mode	The object is merged.

The MDL reads and processes the imported metadata and writes status and diagnostic information in the log file. When the import is complete, the Metadata Import Results dialog displays.

Restoring a Project

Restoring a project allows you to recreate metadata within a Warehouse Builder repository from an external location. Warehouse Builder provides a Restore Wizard to assist you in

this process.

Note:

The Archive and Restore utilities will be desupported in the next release of Oracle9i Warehouse Builder.

You must set up your Archive/Restore settings on the Preferences page before you can archive or restore your project. If you attempt to archive or restore without setting these preferences, you get an error.

Differences Between Restore and Import

Archive and Restore are different from Import and Export. [Table 14-4](#) describes the differences between Restore and Import.

Table 14-4 Differences Between Restore and Import

Feature	Restore	Import
Character Set	UTF8	User Configured
Complete Project Replacement	Yes	Does not delete the project; may replace it, depending on MDL mode
Dump Format	MDL	MDL
UniversalID Preservation	Always	User Configured
Name Preservation	Always	User Configured
Log File Name	Generated	Generated and User configured
Mode	Replace	Create/Update/Replace/ Merge

Restoring a Project

Follow these instructions to restore a project.

To restore a project:

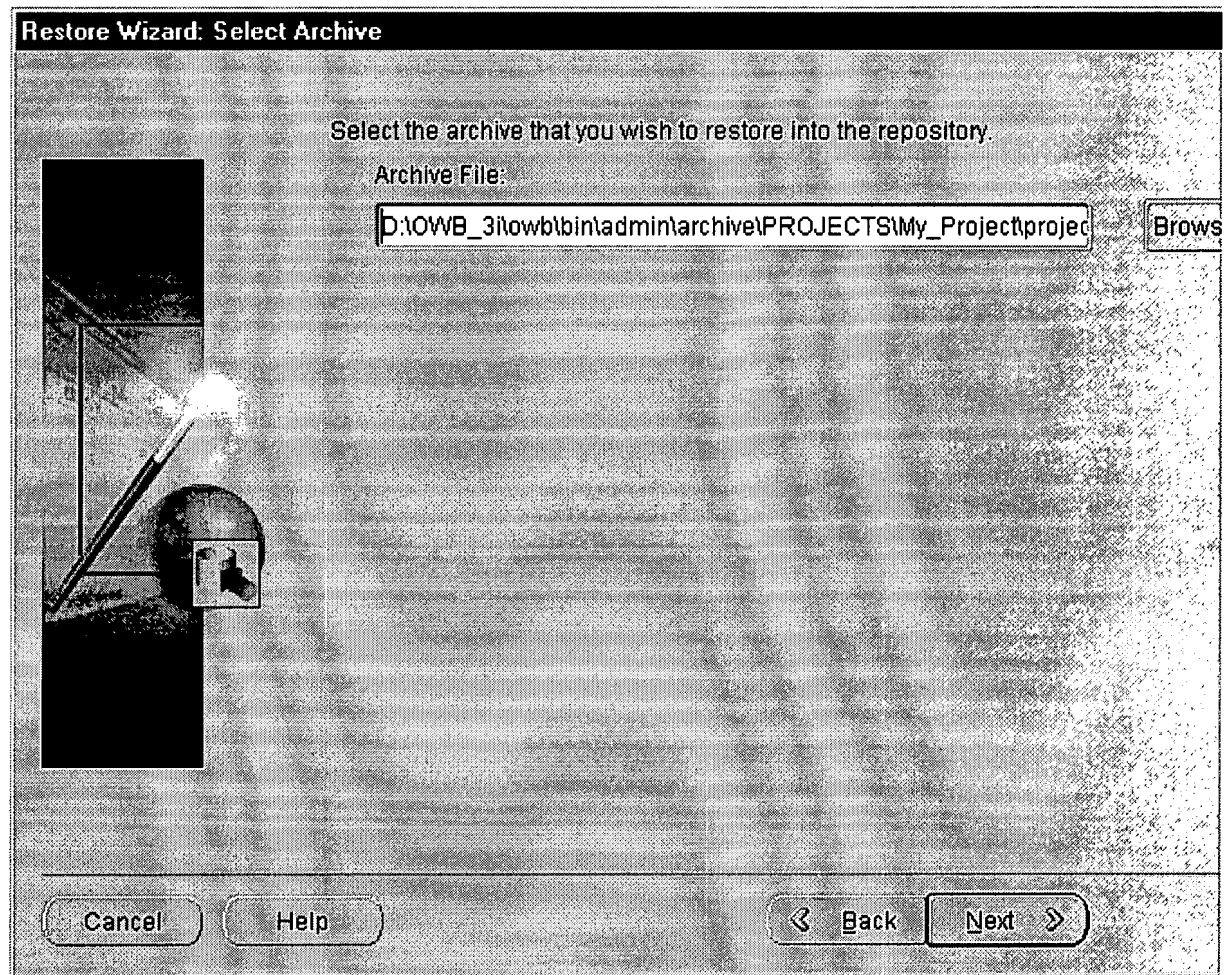
1. Select Restore from the Project menu.

The Restore Wizard Welcome page displays.

2. Click **Next**.

The Select Archive page displays. Browse to or type the Archive File you want to restore.

Figure 14-11 Select Archive Page

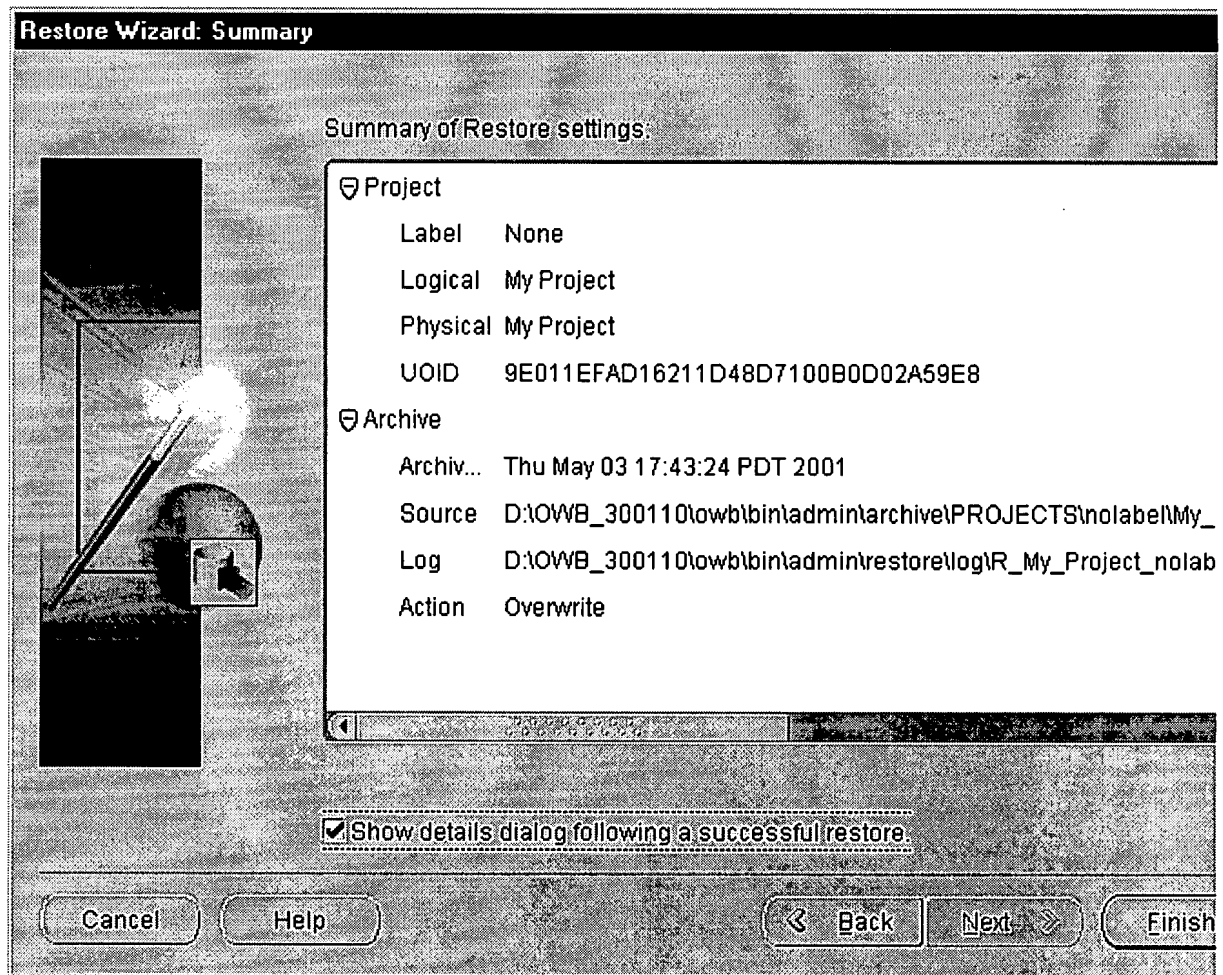


Text description of the illustration rest_wia.gif

3. Click **Next**.

The Summary page displays a summary of the restore settings prior to running the restore process. If you want to see the details of your restore after the restore process is complete, check the Show details dialog following a successful restore box.

Figure 14-12 Restore Wizard Summary Page



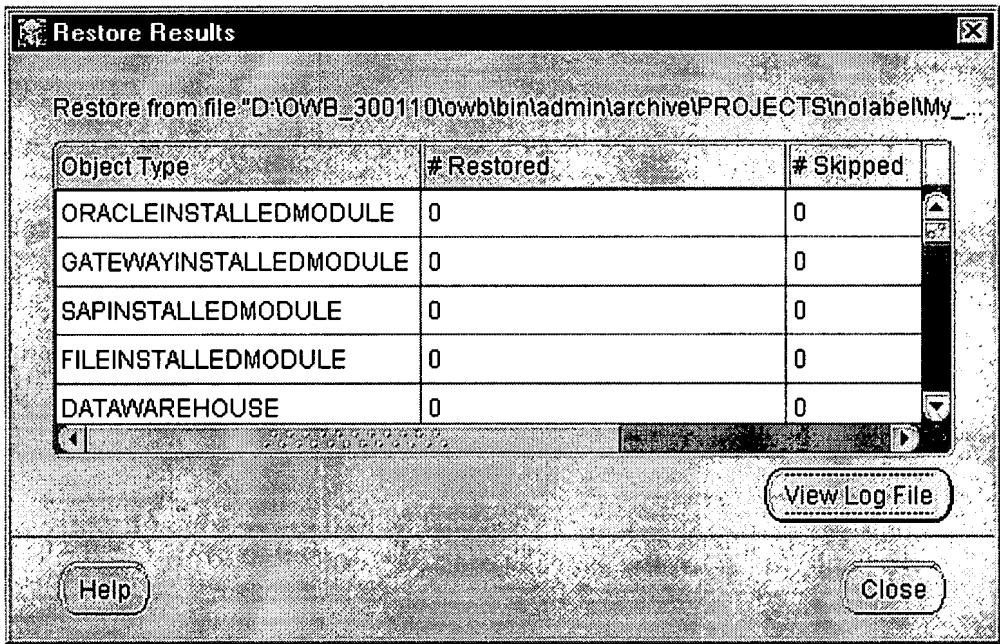
Text description of the illustration rest_wiz.gif

4. Click **Finish**.

This begins the restore process. A progress window appears. When the progress bar reaches 100%, the restore process is complete.

If you checked the Show details dialog following a successful restore box, the Restore Results dialog displays. This dialog displays the name of each object type, how many of each were restored, and how many of each were skipped.

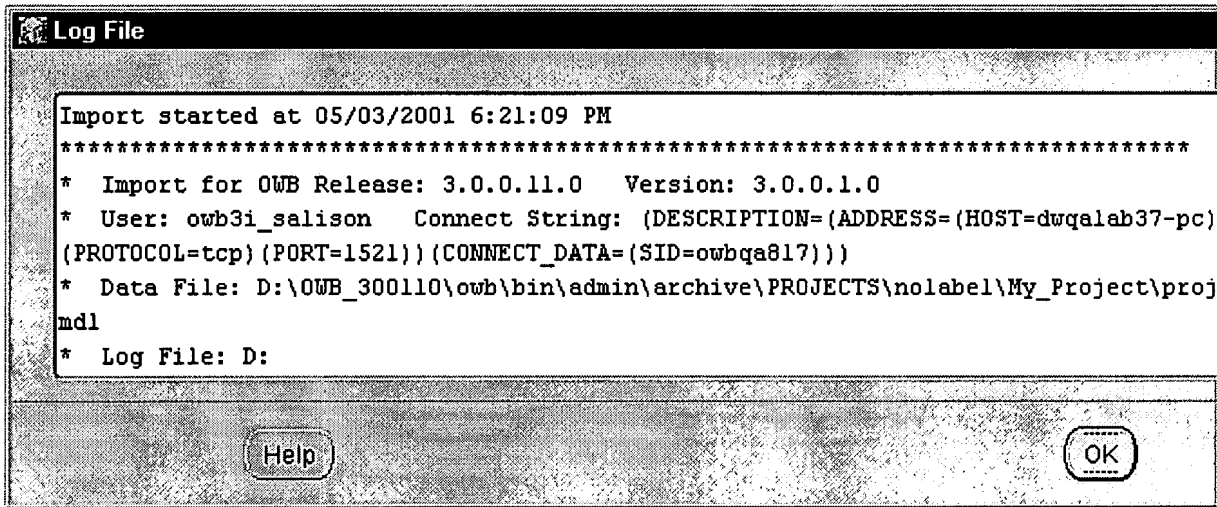
Figure 14-13 Restore Results



Text description of the illustration rest_rsl.gif

For a more detailed look at the archive process, click **View Log File**. This displays the entire log file.

Figure 14-14 Restore Log File



Text description of the illustration rest_log.gif

Using the Metadata Loader Command Line Utility

You can operate the MDL from the command line instead of the user interface.

This section contains the following topics:

- Creating MDL Parameter Files at the Command Line

- [Exporting Metadata Using the Command Line Utility](#)
- [Importing Metadata Using the Command Line Utility](#)

For related information on using the Metadata Loader's command line utility for upgrade purposes, refer to the *Oracle9i Warehouse Builder Installation and Configuration Guide*.

Creating MDL Parameter Files at the Command Line

If you use the command line interface, you can customize how to move your metadata at a more detailed level than you can when using the GUI. For example, when exporting metadata, the command line allows you the flexibility to disable the export of configuration values, vary the separator character within an export file, and maintain parameter files for selected export operations.

For importing metadata, the command line offers the flexibility of creating specific import actions for each object. These operations automate the consolidation or synchronization of metadata in multiple repositories that have a similar project structure.

The scripts for executing both the export and import utilities reside in the `$OWBHOME\owb\bin\win32` directory. Both the export and import utilities are driven by a set of parameters. You can specify the MDL parameters by:

- Providing MDL parameters as a response to command-line prompts.
- Creating an MDL parameters file.
- Providing MDL parameters as a response to command-line prompts, and create a MDL parameters file.

Exporting Metadata Using the Command Line Utility

By default, the MDL also exports values for configuration parameters for loading data, but you can override this setting at the command line. You can choose how you want to export the files within a project. For example, if you are exporting three source modules and two target modules, you can choose to export them separately or together.

To export a project at the command line:

1. Create the MDL parameters file.
2. Execute the Metadata Export Utility.

The following command invokes the Metadata Export Utility and specifies the above parameters file:

```
w:\owb\bin\win32>exp parfile=e:\MDL\EXP_Directives
Processing ... Export successful.
```

The objects are exported to the file and can be imported into a repository using the

metadata import utility.

Keywords for the Export Utility

An MDL parameters file is a text file that contains a set of parameters for the export utility. The format for an export parameter is:

Keyword=Value

You can also form an export parameters by replacing the value with the wildcard character (*), which matches any string, or with a list of named objects:

Keyword=*

Keyword=(value-1, value-2, ..., -k)

For example, you can specify a set of tables to be exported as:

TABLES=(Customers, Products, Days)

Example 14-3 shows a typical parameters file for importing a module.

Example 14-3 Parameters File format

USERID=GCCWH/GCCWH@dwdoc11-pc:1521:ora816
PROJECT=GCCWarehouse
FILE=e:\MDL\GCCWarehouse-exp-JUL01
FIELDSEPARATOR=|
LOG=e:\MDL\GCCWarehouse-exp-JUL01-LOG
CONFIGPARAM=N

Table 14-5 summarizes the keywords used to form export parameters. You can use the comment indicator (#) to document the scripts. Put the indication in the first column of a record and follow it with text.

Table 14-5 Keywords for Export Utility Parameters

Utility Prompt	Keyword	Description
Username/password@host:port:sid	USERID	Username, password and connection string.
N/A	USERNAME	The user name for accessing Warehouse Builder

Utility Prompt	Keyword	Description
		repository.
N/A	PASSWORD	The password that corresponds to the USERNAME.
N/A	HOST	Machine name for Warehouse Builder repository.
N/A	PORT	Port for Warehouse Builder repository database listener.
N/A	SID	SID for Warehouse Builder repository database.
Project Name	PROJECT	Project name. Wildcard for supported for Project, but used, no other object type keywords can follow. In order to export schema transformation use PROJECT=Global Shared.
Export File	FILE	File name for exported data.

Utility Prompt	Keyword	Description
Field Separator	FIELDSEPARATOR	Field separator, ^ or ~.
Log File	LOG	File name for status and statistics of export.
Parameter File	PARFILE	Parameter file containing keywords.
N/A	CONFIGPARAM	Export configuration values (Y/N). Default is Y.
N/A	TRACE	Debug messages. Options: S - write messages to screen Y - write messages to file B - write messages to screen and a
N/A	TRACEFILE	Trace file name
N/A	PHYSICALNAMES	Use physical names (Y/N) for lookup of objects to be exported. Default is N.

Utility Prompt	Keyword	Description
N/A	CHARACTERSET	The character set to be used for the export data file.
N/A	MODULES	If a wildcard multi-value format is used for MODULES, no other object type keywords can follow. In simple format, this keyword can appear multiple times, directly followed by keywords for any of its object types which can be selected using any format (simple, wildcard, multiple).
N/A	TABLES	N/A
N/A	VIEWS	N/A
N/A	FILES	N/A
N/A	SEQUENCES	N/A
N/A	MATERIALIZED VIEWS	N/A

Utility Prompt	Keyword	Description
N/A	DIMENSIONS	N/A
N/A	FACTS	N/A
N/A	TRANSFORM CATEGORIES	For wildcard multi-value format, no FUNCTION keyword can follow. If similar format then keyword can appear multiple times, directly followed by FUNCTION keyword, we can use any format (similar, wildcard, multiple).
N/A	FUNCTIONS	N/A
N/A	MAPPINGS	N/A
N/A	COLLECTIONS	N/A
N/A	LOCATIONS	N/A
N/A	CONNECTORS	N/A
N/A	RUNTIMEREPOSITORYCONNECTIONS	N/A
N/A	STANDALONEFUNCTIONS	N/A

Utility Prompt	Keyword	Description
N/A	STANDALONEPROCEDURES	N/A
N/A	ADVANCEDQUEUES	N/A
N/A	EXTERNALTABLES	N/A
N/A	PROCESSES	N/A
N/A	SNAPSHOTS	N/A
N/A	QUERYOBJECTS	N/A
N/A	REPORTS	N/A
N/A	REPORTGROUPS	N/A
N/A	IOBUSINESSAREAS	N/A
N/A	HELP	Use HELP= for a complete list.
N/A	#	Comment line used in a parameter file

Importing Metadata Using the Command Line Utility

To import selected modules:

1. Create an MDL parameter file.
2. Execute the Metadata import utility.

The following command invokes the Import Utility and specifies the above MDL parameter file:

```
w:\owb\bin\win32>imp parfile=e:\MDL\IMP_Directives.txt
Processing ...
Import successful.
```

Keywords for the Import Utility

Like the MDL export, you can direct the MDL import to import objects from a file by answering prompts or by creating a file with a set of parameters. [Example 14-4](#) shows a typical parameters file for importing a module.

Example 14-4 Parameters File format

```
USERID=GCCWH/GCCWH@dwdoc11-pc:1521:ora816
FILE=e:\MDL\gccstar-exp
LOG=e:\MDL\gccstar-imp-LOG
MODE=CREATE
CONFIGPARAM=N
```

[Table 14-6](#) summaries the keywords used to form import parameters.

Table 14-6 Keywords for Import Utility Parameters

Utility Prompt	Keyword	Description
Username/passw@host:port:sid	USERID	Username, password and connection as a string.
N/A	USERNAME	The user name for accessing Warehouse Builder repository.
N/A	PASSWORD	The user password that matches USERNAME.
N/A	HOST	Machine name for Warehouse Builder repository.
N/A	PORT	Port for Warehouse Builder repository.
N/A	SID	SID for Warehouse Builder repository.

Utility Prompt	Keyword	Description
Import File	FILE	File name for the data to be imported.
Import Mode	MODE	CREATE, REPLACE, UPDATE, or INCREMENTALUPDATE.
Log File	LOG	File name for the status and statistics of the export.
Parameter File	PARFILE	Parameter file containing keywords.
N/A	CONFIGPARAM	Import configuration values (Y/N). Default is Y.
N/A	TRACE	Debug messages. Options: S - write messages to screen Y - write messages to a file B - write messages to screen and a file
N/A	TRACEFILE	Trace file name.
N/A	PHYSICALNAMES	Use physical names (Y/N) to lookup objects to be imported. Default is Y.
N/A	CHARACTERSET	The character set to use for the export data file.
N/A	HELP	Use HELP=Y for a complete list.

Utility Prompt	Keyword	Description
N/A	#	Comment line used in a parameter file.
N/A	IGNOREUniversalID	Ignore (Y/N) the universal id as the search criteria. Default is N.
N/A	PRESERVEDESCRIPTION	Preserve the description (Y/N) of already existing objects if the MDL data file does not have a description for the object. Default is N.
N/A	SINGLEUSER	Request a single user lock (Y/N) for running the import. Default is N.

If a MODE parameter is not included, then the default is CREATE.

In addition to running an MDL parameter file from the import utility, you can also specify an action plan within the file that will allow you to specifically define what you want to do with each object in the imported file. First you need to specify if you want the object imported, skipped, or deleted. If you choose to import the object, you can set the import mode to CREATE, UPDATE, REPLACE, or INCREMENTAL UPDATE.

Example 14-5 shows an example of an MDL parameter file that contains an action plan.

Example 14-5 MDL Action Plan

```

USERID=user_sample/user_sample@test-pc:1521:ora8i
#
FILE=e:\test\data\sample_file.mdl
LOG=e:\test\log \imp_sample_file.log
#
MODE=ACTIONPLAN
PHYSICALNAMES=Y
IGNOREUOID=Y
#
# User-Specified Action Plan
#
ACTION=NONE
PROJECT=MY PROJECT
MODULES=(DATAWAREHOUSE)
#
ACTION=CREATE
TABLES=(TABLE_3)
FACTS=(FACT1, FACT2, FACT3)

```

```

SEQUENCES=(SEQ_A, SEQ_B, SEQ_C)
#
ACTION=REPLACE
TABLES=(TABLE_1, TABLE_2)
DIMENSIONS=(DIM1, DIM2, DIM3)
#
ACTION=DELETE
TABLES=(TABLE_A, TABLE_B)
#
# Switching to a different module
ACTION=REPLACE
MODULES=(FLAT_FILE)
FILES=(FILE_1, FILE_2)
#
ACTION=CREATE
FILES=(FILE_3)
#
ACTION=DELETE
FILES=(FILE_X)

```

Splitter for Exporting and Importing Warehouse Builder Mappings

The Split utility provides a workaround for the memory limitations of the MDL import utility when you are importing a large number of mappings. This utility generates export and import scripts for migrating mappings in pieces as opposed to migrating them all at the same time. The generated scripts have matching MDL parameter files that utilize the CREATE mode. These files can be edited.

If the MDL import fails because of large data, the split utility can be used to re-export and import the mapping data in smaller pieces. All other object types must be exported and imported using the standard MDL utilities. Only mappings can be split into smaller pieces. To export all entities other than mappings, a parameter file containing the following can be used:

- VIEWS=*
- TABLES=*
- SEQUENCES=*
- MATERIALIZEDVIEWS=*
- CUBES=*
- FILES=*
- DIMENSIONS=*
- VIRTUALTABLES=*
- TEMPORARYTABLES=*
- TRANSFORMCATEGORIES=*

The `split` utility splits the mappings within a module in a Warehouse Builder project. The size of the pieces is determined by a parameter located in a file provided with this application.

The `expsplit` batch script accepts the following arguments:

- A parameter file, specified in the form `c:\temp\owb_apps.txt`. The parameter file has special keywords outlined below that identify the number of mappings, data file names, and extensions.
- A parameter target file prefix, specified in the form `c:\temp\owb_apps` the piece number (numbered from 1). A `.txt` suffix is added to the generated parameter file, a `.bat` suffix is added for the export batch file, and an `_imp.bat` suffix is added to the import batch file.

The following is an example of how to start the `split` utility:

```
expsplit exampleparams.txt c:\temp\ora_apps
```

The following example uses a parameter file `exampleparams.txt`. This file contains the following parameters:

- `userid=apps/apps@130.35.12.73:1521:orcl0`
- `PHYSICALNAMES=Y`
- `LOG=c:\temp\owb_data_apps`
- `LOGEXT=log`
- `FILE=c:\temp\owb_data_apps`
- `FILEEXT=dat`
- `FIELDSEPARATOR=^`
- `PROJECT=EDWPRJ`
- `MODULES=EDW_COMMON_MODULE`
- `TYPE=MAPPINGS`
- `COUNT=70`

This file is similar to the export parameter file for Warehouse Builder Metadata Loader, with the changes listed in [Table 14-7](#).

Table 14-7 Split Utility Export Parameter Keyword Descriptions

Keyword in	Description
------------	-------------

Parameter File

FILE	Prefix of data file, the chunk number, and file extension (FILEEXT) define the data file name where you exported the data.
FILEEXT	The data file extension.
PHYSICALNAMES	Used for name matching.
LOG	Prefix of the log file, the chunk number, and file extension (FILEEXT) define the log file name.
LOGEXT	The log file extension.
PROJECT	A single project name must be specified.
MODULES	A single module name must be specified.
TYPE	Must be MAPPINGS.
COUNT	The number of mappings to be written to each export chunk.

If the mappings for a Warehouse Builder project are split, the generated parameter files are named as follows:

```
owb_apps1.txt
owb_apps2.txt
```

A batch file is generated: `c:\temp\owb_apps.bat` (given the parameter target file prefix) to export the data from the repository. An import batch file is created to import, using create mode, into the same repository. These files can be edited if different target databases are required.

To migrate data using the split utility:

1. Using command line or Warehouse Builder, perform MDL export of all objects other than mappings.

To export all objects other than mappings in a command line, use a parameter file with the following keywords:

VIEWS=*

TABLES=*

SEQUENCES=*

MATERIALIZEDVIEWS=*

FACTS=*

FILES=*

DIMENSIONS=*

VIRTUALTABLES=*

TEMPORARYTABLES=*

TRANSFORMCATEGORIES=*

If you are using Warehouse Builder to perform export, use multi-select to select and export objects other than mappings.

2. Import the new export file into target repository.
3. Split the mappings and export them using split utility

```
expsplit exampleparams.txt c:\temp\ora_apps
```

The utility connects to the source repository, splits mappings, and creates multiple parameter files according to `exampleparams.txt`. These parameter files are used during the export. The utility also creates an export batch file and an import batch file.

Table 14-8 lists the files that are created.

Table 14-8 Files Created by the Split Utility

Description	File Name
Batch file to perform export	c:\temp\ora_apps.bat
Batch file to perform import	c:\temp\ora_apps_imp.bat
4. Run the export batch file to export mappings into the location specified in the parameter file (variable FILE specified in step 3).	c:\temp\ora_apps1.txt c:\temp\ora_apps2.txt
5. Modify generated parameter files c:\temp\ora_apps1.txt, c:\temp\ora_apps2.txt. Edit the connection information to point to the target repository.	c:\temp\ora_apps3.txt

6. Run import batch file c:\temp\ora_apps_imp.bat to complete the import.

[Go to previous
page](#)

[Go to next
page](#)

[Oracle](#)
Copyright © 1996, 2003 Oracle Corporation.
All Rights Reserved.

[Go To Table
Of Contents](#)
[Contents](#)

[Go To
Index](#)
[Index](#)

[Skip Headers](#)

Oracle9i Warehouse Builder User's Guide
Release 2 (9.0.4)
Part Number B10657-01

Go To Table	Go To
Of Contents	Index
Contents	Index

[Go to previous page](#) [Go to next page](#)

17

Managing Warehouse Builder Browser

Warehouse Builder Browser integrates with Oracle Portal to enable you to add and customize Warehouse Builder portlets. The Warehouse Builder Browser uses Warehouse Builder Public Views to create pre-built reports on all repository objects and relationships between objects. You can also use the Public Views to create your own custom reports.

This chapter contains the following information:

- [Warehouse Builder Browser Overview](#)
- [Adding Portlets](#)
- [Administering Warehouse Builder Browser](#)
- [Creating Custom Reports](#)

Warehouse Builder Browser Overview

Warehouse Builder Browser is a web based application that you can use to extend, access, and run reports on your repository metadata. To view the metadata and access these reports, you must have access to Oracle9iAS Portal.

When you first install or upgrade Warehouse Builder, you can use the Warehouse Builder Browser Assistant available on the start menu to install the Warehouse Builder Browser. The Browser Assistant enables you to set up an Oracle9iAS Portal from which you can access and create metadata reports. You can run the Browser Assistant during your installation process or defer it to a later time.

After Warehouse Builder Browser is installed, you can add the Warehouse Builder portlets to your Portal home page. You can run Metadata reports using the Browser from either the Warehouse Builder client or Oracle Portal.

About Oracle9iAS Portal

Oracle9iAS Portal enables you to create and view database objects using an HTML-based interface. It provides tools for creating HTML-based interfaces. Portal provides you with a centralized and personalized view of relevant applications and data in a single web site.

The fundamental building blocks of an Oracle9iAS Portal site are called portlets. A portlet is a re-usable information component that summarizes or provides access to an information source. Portlets can stand alone in a Portal site, link to other portlets or portal sites, or they can be nested within one another.

Warehouse Builder Browser integrates with Oracle9iAS Portal to obtain metadata reporting portlets.

Portlets for Managing Warehouse Builder Browser

The Warehouse Builder Browser consists of portlets that you can add to customize your portal page. You can add more than one of each portlet. For example, you can add two Reports portlets with each running a report on a different repository.

The portlets used for managing Warehouse Builder Browser include the following:

- [Launcher Portlet](#)
- [Administration Portlet](#)
- [Reports Portlet](#)

To access any portlet, you must first configure the Warehouse Builder Browser to access a Warehouse Builder Design Repository.

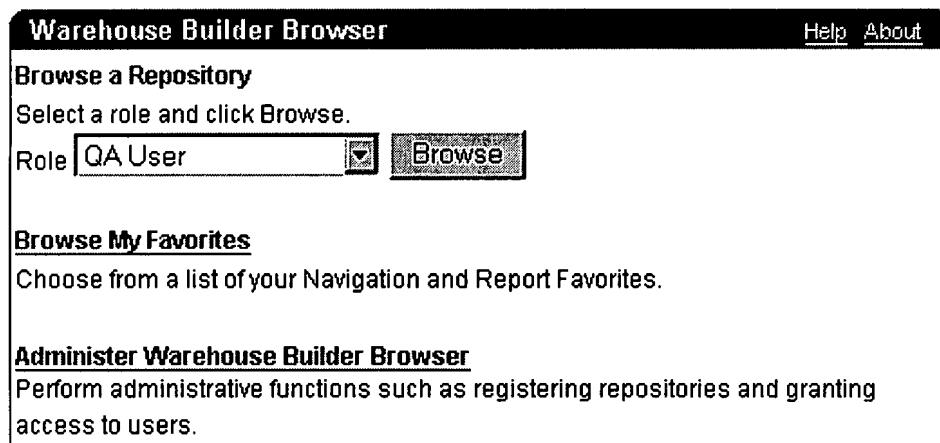
You can also add custom portlets to suit your business intelligence needs. For more information, see ["Adding Portlets"](#).

Launcher Portlet

The Launcher Portlet provides access to all the available functions. Other portlets contain subsets of all available functions.

- **Browse a Repository:** Select a role and click **Browse** to browse the repositories available to the current user. The main Warehouse Builder Browser page displays in full page mode. You can select from a list of repositories, and use the Navigator to view the detail of that repository.
- **Browse My Favorites:** Select this link to view your Warehouse Builder Favorites in full page mode.
- **Administer Warehouse Builder Browser:** Select this link to view the Warehouse Builder Administration in full page mode. Use these pages to configure your Browser.

Figure 17-1 Launcher Portlet



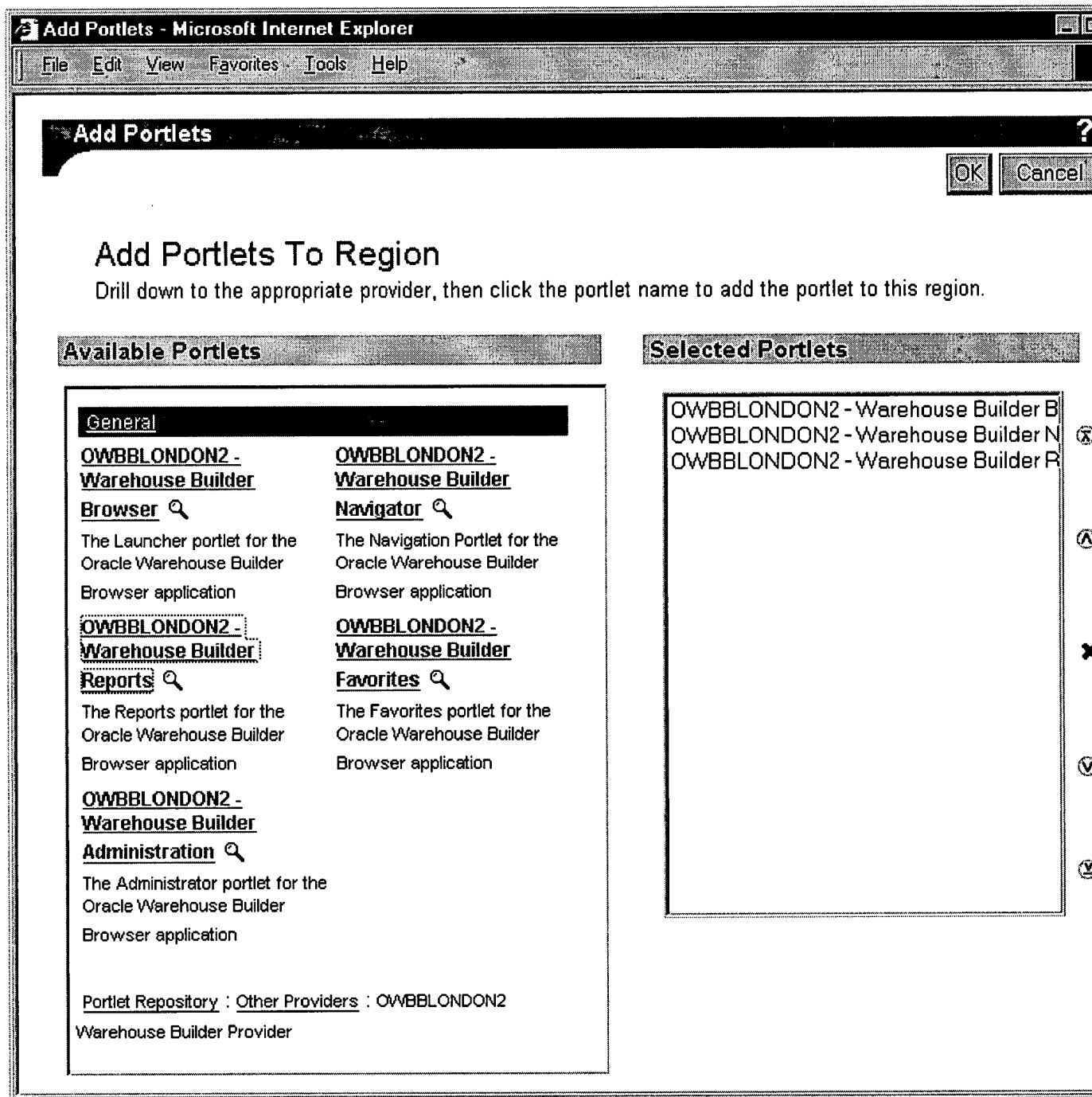
Text description of the illustration laun_poa.gif

Administration Portlet

The Administration Portlet provides access to the following Administration features from your Oracle Portal home page:

- **Register an OWB Repository:** Register Warehouse Builder repositories and maintain database links.
- **Register a Custom Report:** Register custom reports.
- **Purge Stale User Information:** Purge obsolete Warehouse Builder Browser settings.
- **Resource Management:** Manage access privileges to Warehouse Builder Browser resources.
- **Manage Preferences:** Save and load preference settings from files or an existing schema.
- **Manage Dependency Index:** Improve performance of Impact Analysis by specifying how often to refresh the Dependency Index.

Figure 17-2 Administration Portlet



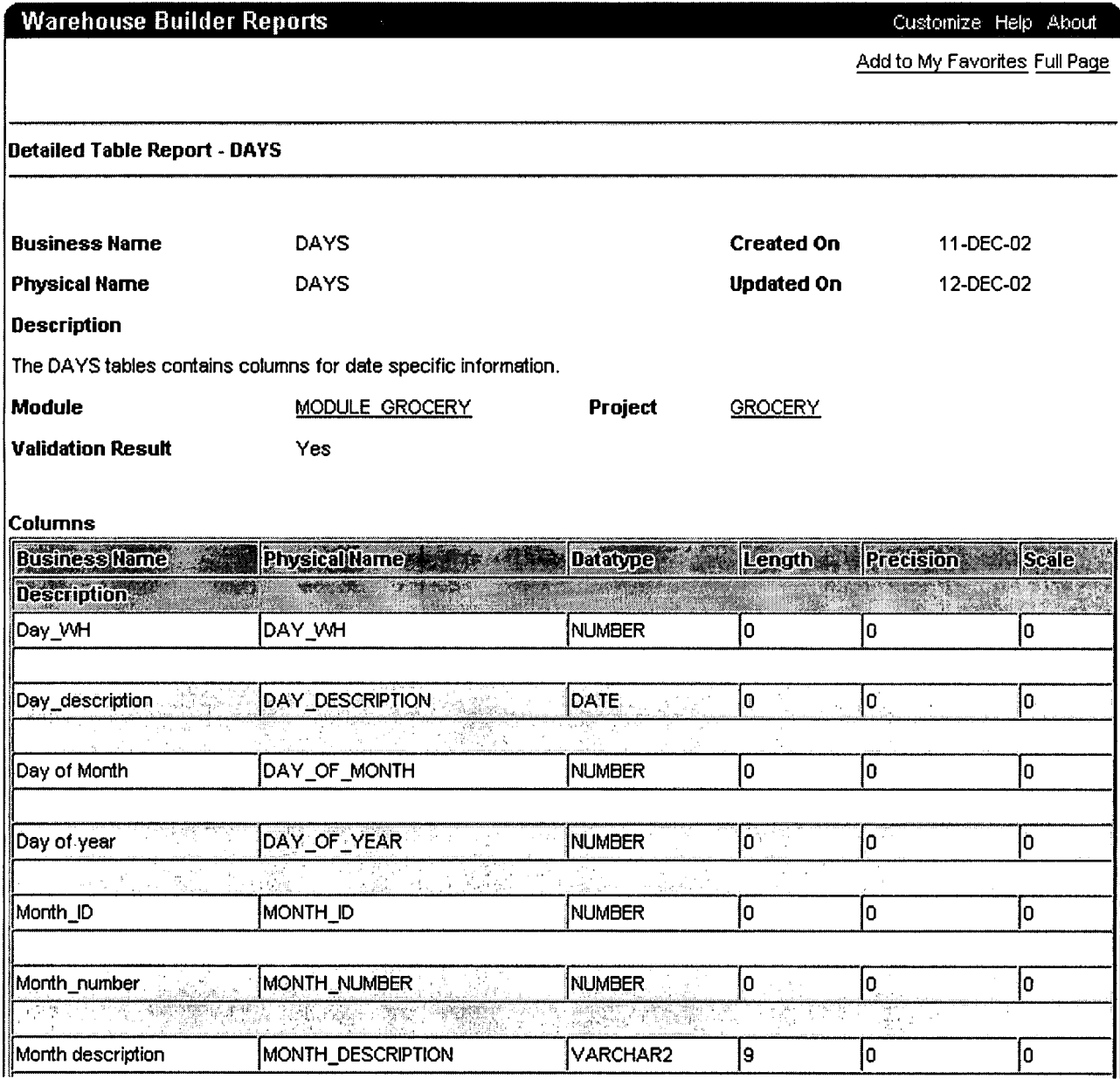
Text description of the illustration addport.gif

Reports Portlet

The Reports Portlet displays a Warehouse Builder Report on your Oracle Portal home page. The Reports Portlet displays one of your favorite reports. When you first add this portlet to a page, it does not contain a default report. You must first use the customize option to select and add a report from you favorites list.

When you have a Reports portlet on your Oracle Portal home page, it refreshes each time you reload the home page. This can delay the home page display.

Figure 17-3 Reports Portlet



[Text description of the illustration 13_5a.gif](#)

Adding Portlets

Warehouse Builder Browser portlets can be added to Oracle Portal after they have been installed on the machine running Oracle Portal.

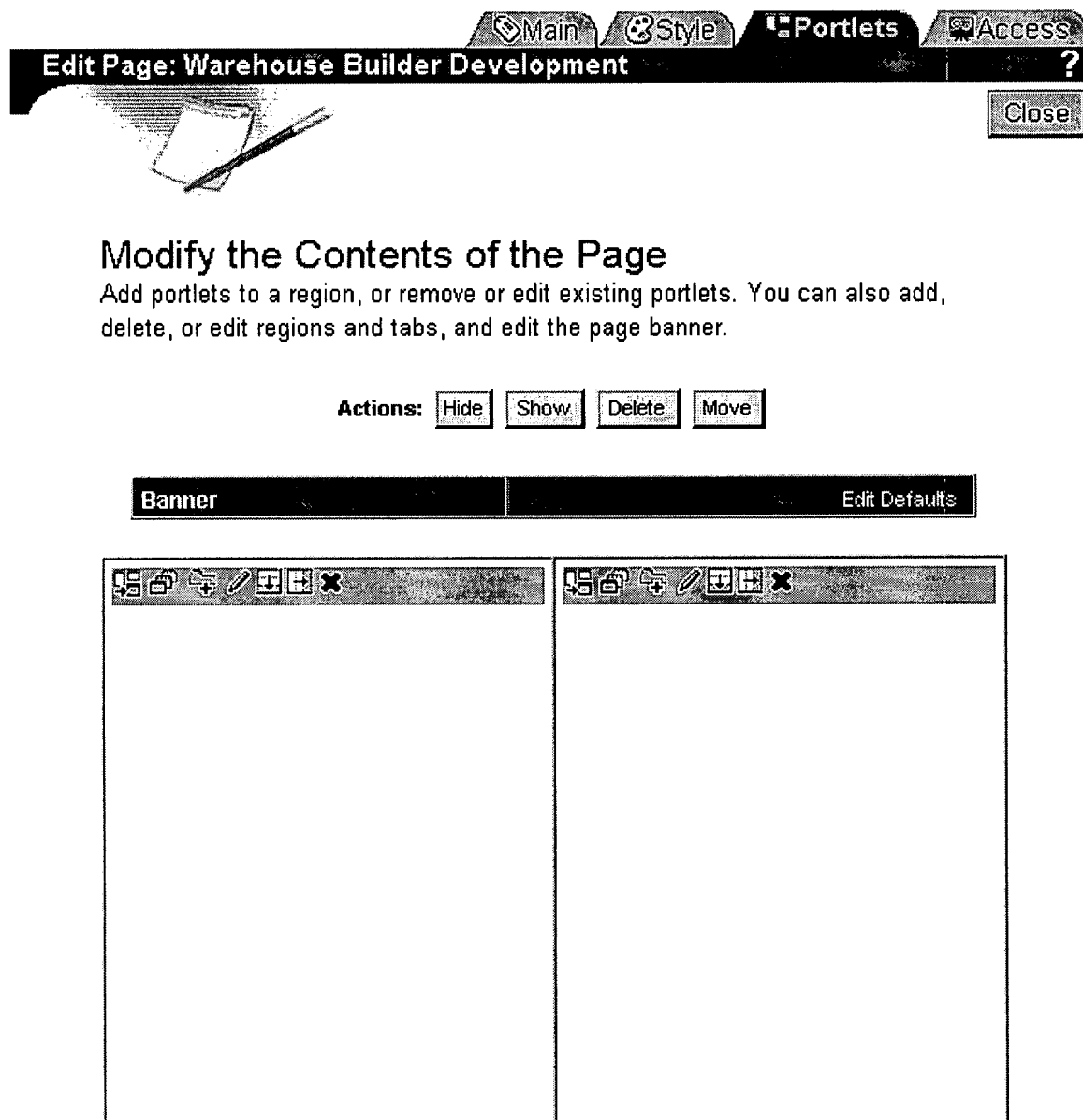
For information about customizing Oracle Portal pages, see the Oracle Portal documentation.

To add a portlet to a page:

1. From the Oracle Portal page, select the Edit Page link from the upper-right corner of the page.

The Edit Page displays the contents of the Oracle Portal page.

Figure 17-4 Oracle Portal Edit Page

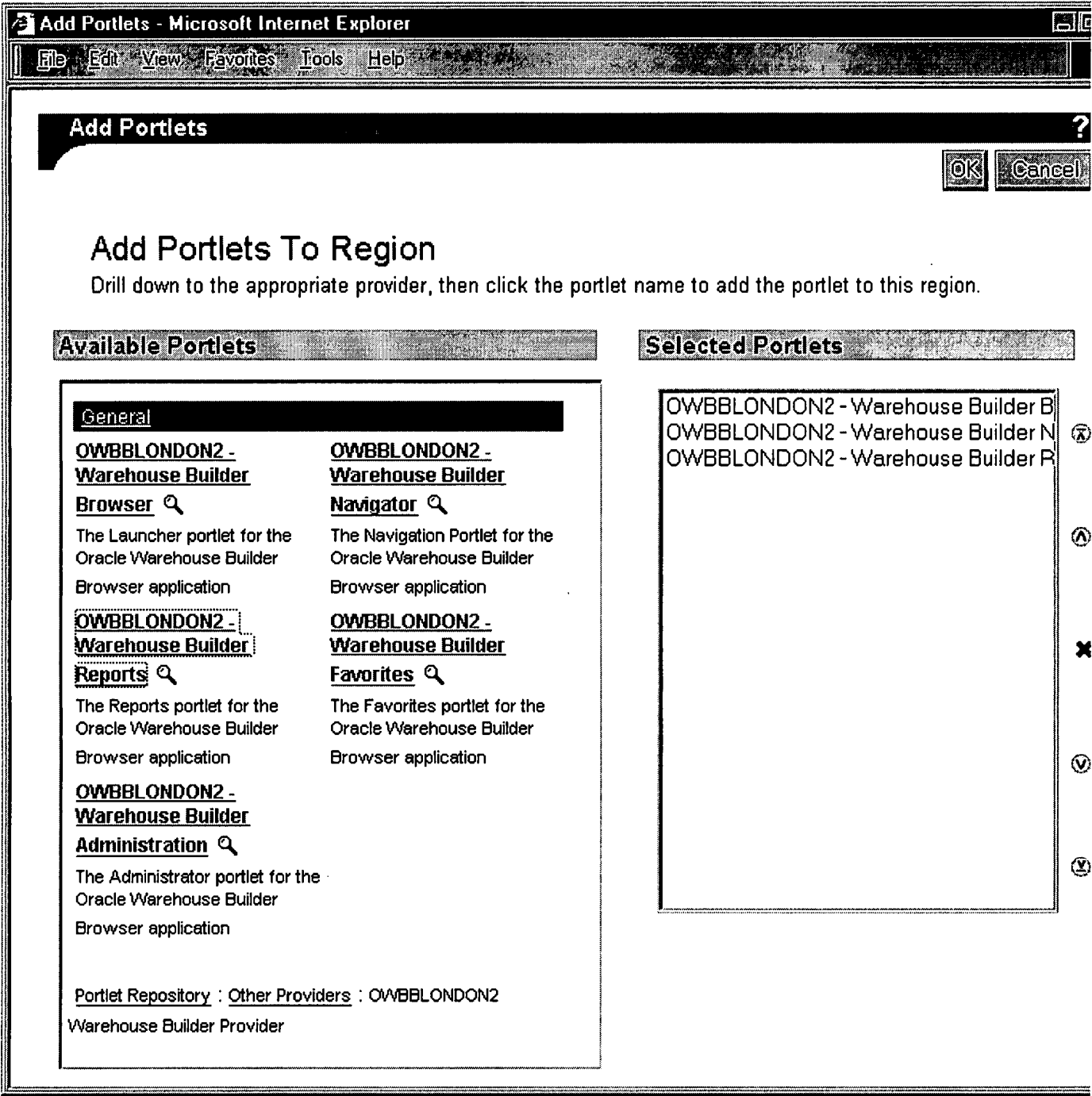


Text description of the illustration 14_6.gif

2. Select the Add Portlets icon.

The Add Portlets page displays a list of available portlets on the left side, and a list of selected portlets on the right side.

Figure 17-5 Add Portlets Page



Text description of the illustration addport.gif

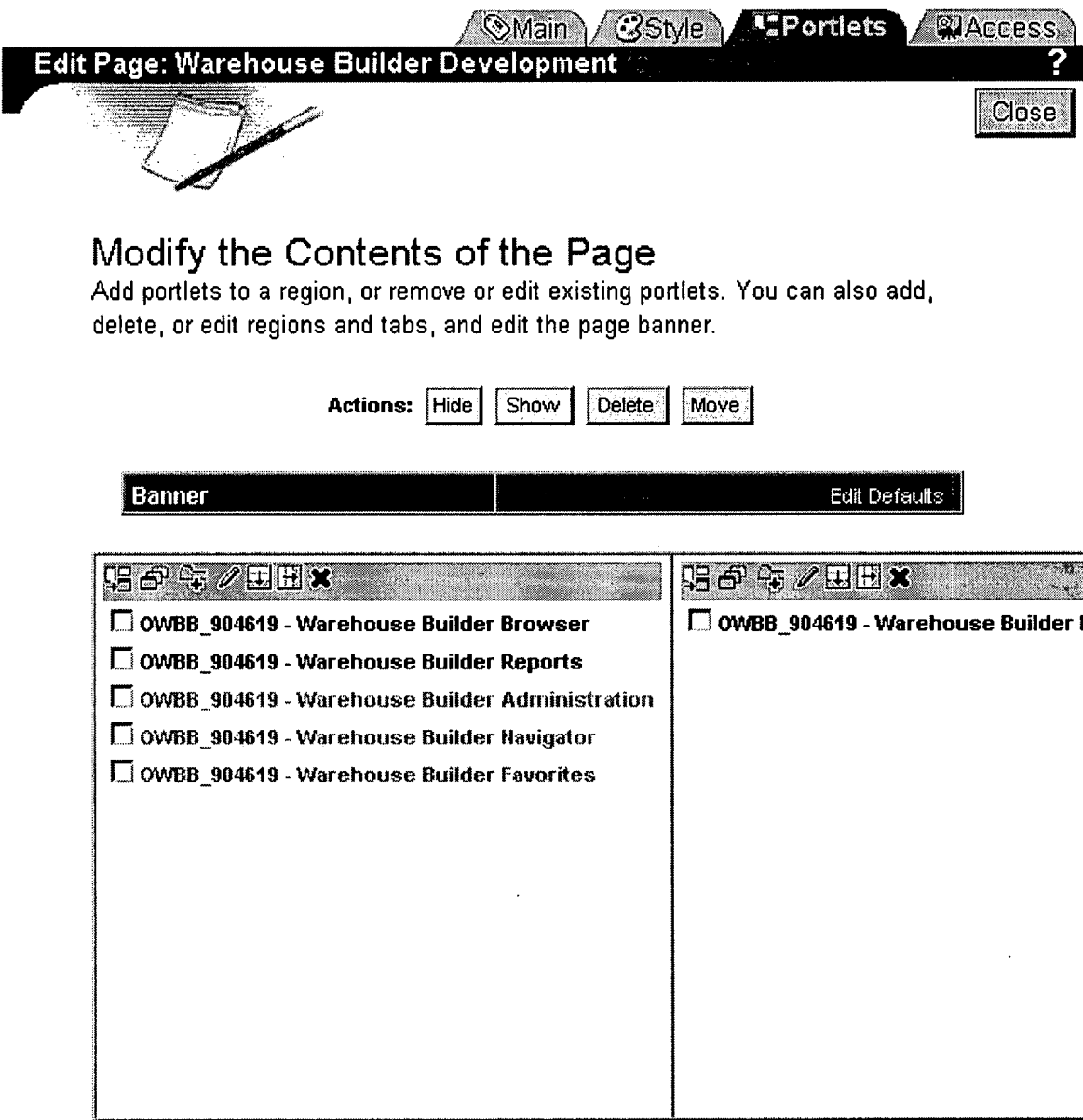
3. Select the portlets to add to the Oracle Portal home page.

The portlets you have added display in the right column. You can organize them by using the arrow buttons and delete them by using the X button.

4. Click **OK** when you are done.

The Edit Page displays the Warehouse Builder portlets that you added.

Figure 17-6 Edit Page



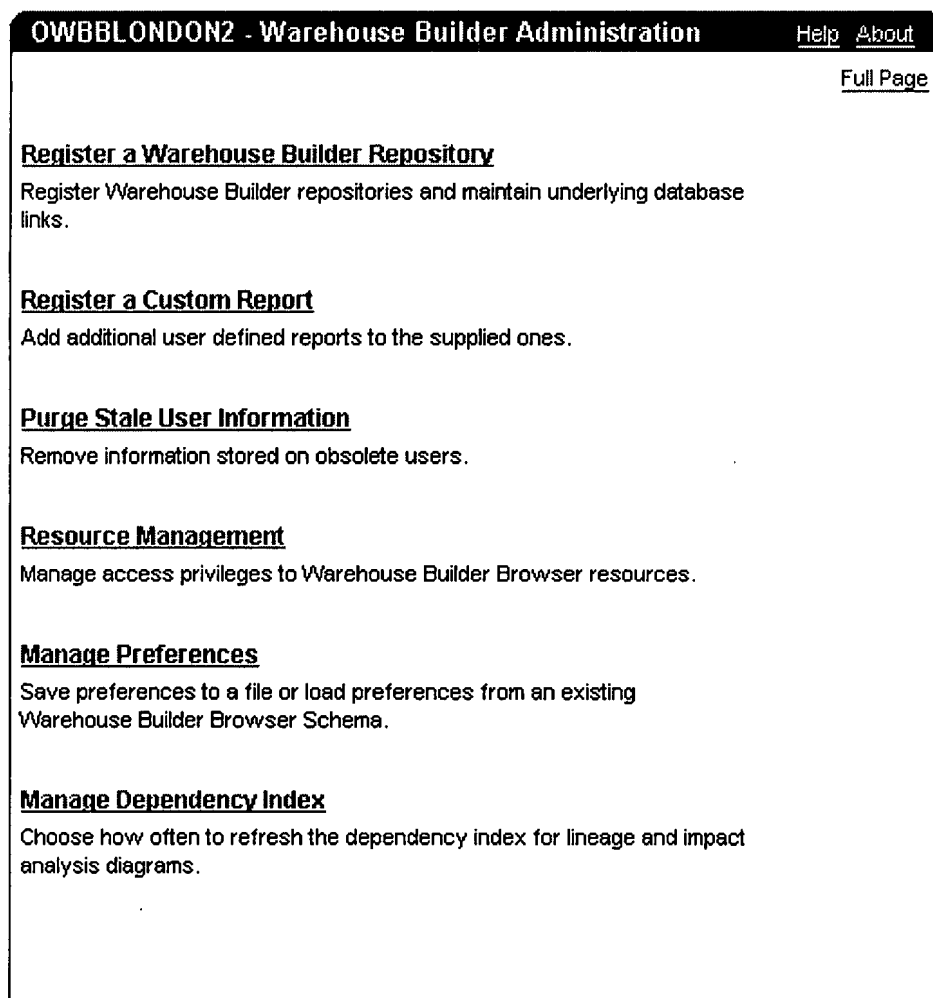
Text description of the illustration 14 8.gif

Administering Warehouse Builder Browser

The Warehouse Builder Administration pages can only be accessed by Oracle Portal users with full administrator privileges.

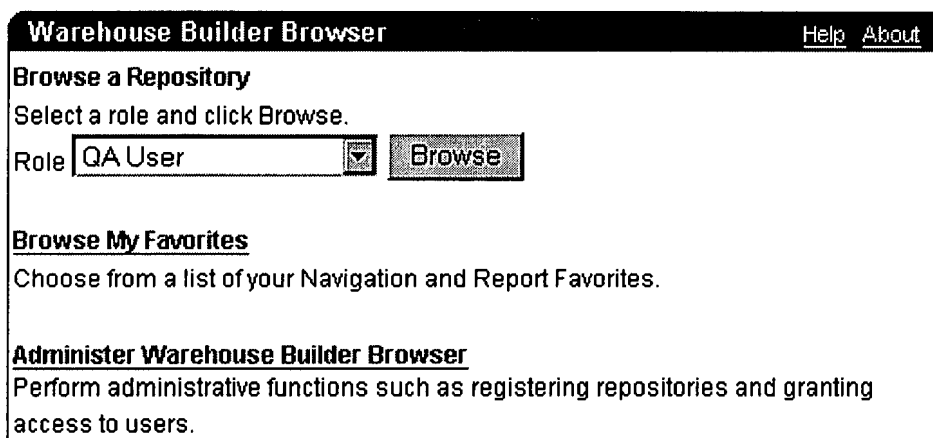
From the Oracle Portal Home page, select the Administer Warehouse Builder Browser link to access the Warehouse Builder Administration pages.

Figure 17-7 Administration Portlet in Warehouse Builder Browser



Text description of the illustration admn_por.gif

Figure 17-8 Launcher Portlet in Warehouse Builder Browser



Text description of the illustration laun_poa.gif

The Administration page contains links for the following administration actions:

- Register a Warehouse Builder Repository

- [Register a Custom Report](#)
- [Resource Management](#)
- [Managing Preferences](#)
- [Managing the Dependency Index](#)

Register a Warehouse Builder Repository


Before you can access repository metadata reports, register the Warehouse Builder repository with the Warehouse Builder Browser.

To register a Warehouse Builder Design Repository:

- 2
1. Click **Register an OWB Repository** on the Warehouse Builder Administration home page. The Register Repository page displays.

Figure 17-9 Register Repository Page

Register Repository



Apply

OK

Cancel

Administer Database Links

Create, edit or drop database links to Warehouse Builder repositories.

Warehouse Builder Repository Properties

Enter or update the properties of a Warehouse Builder repository.

Name

Database Link

Description

Text description of the illustration 14_11.gif

2.
- Specify the Warehouse Builder repository properties. Table 17-1 lists the properties.

Table 17-1 Warehouse Builder Repository Properties

Field	Description
Name	The user-defined name to identify the repository in the browser system. This name

Field	Description
	is displayed in the navigation pages.
Database Link	The name of the database link used to access the repository. The link must already be created using the Administer Database Links page. This field must be specified even if the repository is in the same database as the browser system.
Description	The user-defined descriptive text. This appears in the navigation pages for the repository.

3. Click **Apply** to register the repository.

4. Click **OK**.

The repository displays in the Warehouse Builder Administration home page.

Managing Repositories

The Resource Management page lists all registered repositories. The actions listed next to the repository are described in [Table 17-2](#).

Table 17-2 Repository Management

Actions	Description
Access	Use this to grant or revoke repository access privileges to the users.
Edit	Use this to edit repository properties.
Unregister	Use this to unregister the repository. After unregistering the repository, it can no longer be browsed using the Browser system. You must re-register the repository if you want to browse it again.

Creating Database Links

Use the Administer Database Links page to connect to Warehouse Builder repositories from the Browser.

Figure 17-10 Administer Database Links Page



Create Database Link

Existing Database Links

A database link with an invalid status indicates either the database link is not functioning or the schema it points to is not a Warehouse Builder repository schema

Database Link	Public View Version	Owner	Status	Created On	Actions
MYREPOSITORY_US_ORACLE.COM	9.0.4	OWBB_904619	Valid	12-DEC-02	edit drop

Text description of the illustration 14_13.gif

To create a database link:

1. Click **Register an OWB Repository** on the Warehouse Builder Administration home page. The Register Repository page displays.
2. Click the link for Administer Database Links.
3. Select **Create Database Link** from the Administer Database Links page.

Figure 17-11 Create Database Links Page

Create Database Link

Database Link Name

Warehouse Builder Repository User Name

Warehouse Builder Repository Password

Remote Database Information
Specify the host address, service name, protocol and host port number.

Host Address

Host Service Name

Host Protocol

Host Port Number

Text description of the illustration 14_14.gif

4. Specify the database link name.
5. Specify the Warehouse Builder Design repository user name and password.
6. Specify the remote database information.
Provide the host address, service name, protocol, and host port number.
7. Click **Apply** to connect the link.
8. Click **OK**.

The new link displays on the Administer Database Links page.

Viewing a Database Link

To view a database link:

1. Select the name of the database link from the Administer Database Links page.
The View Database Link page displays with a detailed report on the database link you selected.
2. Click **OK**.

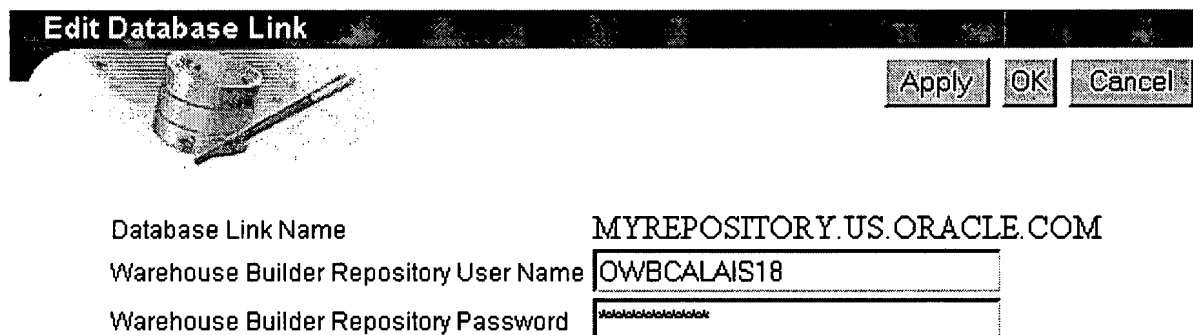
The browser returns to the Administer Database Links page.

Editing a Database Link

To edit a database link:

1. From the Administer Database Links page, select **edit** for the database link you want to alter. The edit link is located under the Actions column.
The Edit Database Link page displays.

Figure 17-12 Edit Database Link Page



Edit Database Link

Database Link Name MYREPOSITORY.US.ORACLE.COM

Warehouse Builder Repository User Name OWBCALAIS18

Warehouse Builder Repository Password XXXXXXXXXXXX

Remote Database Information

Enter a remote database alias OR enter the parameters that are necessary for connecting to the remote database in the form of:

(DESCRIPTION = (ADDRESS = (PROTOCOL = <protocol>) (Host = <hostname>) (Port = <portno>)))
 (CONNECT_DATA = (SERVICE_NAME = <remote Database service name>)))

```
(DESCRIPTION = ( ADDRESS_LIST = ( ADDRESS = ( PROTOCOL =
TCP ) ( HOST = 130.35.50.1 ) ( PORT = 1521 ) ) ) (
CONNECT_DATA = ( service_name = ags.us.oracle.com ) ) )
```

Text description of the illustration 14_15.gif

2. Edit the database link and click **Apply**.
3. Click **OK**.

Dropping a Database Link

Dropping a database link deletes it permanently. You must create a new link to use it again.

To drop a database link:

1. If the database link has been used to register Warehouse Builder repositories, unregister the Warehouse Builder repositories.
2. From the Administer Database Links page, select **drop** for the database link you want to drop. The drop link is under the Actions column.

The database link is dropped and the browser returns to the Administer Database Links page.

Unregistering a Repository

To unregister a Repository:

1. Select the Administer Warehouse Builder Browser link from the Browser page.
2. Select **Manage Resources**.

The Warehouse Builder Administration page displays as shown in [Figure 17-13](#). The table at the bottom of the page lists the registered repositories listed.

Figure 17-13 Registered Repositories and Roles

Resource	Type	Actions
Launcher Portlet	Portlet	access
Apps	Repository	access edit unregister
Warehouse Builder	Repository	access edit unregister
QA User	Role	access
Warehouse Engineer	Role	access
Warehouse User	Role	access

Text description of the illustration unreg.gif

3. Select the repository to unregister and click on the **unregister** link.

The repository is unregistered and no longer appears in the list of registered repositories. You can no longer browse it using the Browser.

Register a Custom Report

A custom report is an application component created in a tool such as the Oracle Portal facilities. Registration of the report provides the browser system with the information required to invoke the report. For more information, see ["Creating Custom Reports"](#).

To register a custom report:

1. From the launcher portlet, click the **Administer Warehouse Builder Browser** link, and then select the **Register a Custom Report** link.

The Register a Custom Report page displays.

Figure 17-14 Register Custom Report Page

Warehouse Builder Administration - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address Links

ORACLE

Warehouse Builder Browser

Home Help

Register a Custom Report

Apply OK Cancel

Custom Report Properties

Enter or update the properties of a custom report.

Display Name

Type name

Package

Repository

Copyright© 2001, Oracle Corporation. All Rights Reserved

Done Local intranet

Text description of the illustration 14 16.gif

2. Enter a display name for the report.
3. Select the type and repository from the drop-down lists. [Table 17-3](#) lists the custom report properties.

Table 17-3 Custom Report Properties

Field	Description
Display Name	Name of the report. This name is displayed on the Reports List page.
Type Name	Name of the data type reported on by this report.
Package	The full name of the PL/SQL package which implements this report.

Field	Description
Repository	Name of the repository containing target objects for this report.
<hr/>	
4. Enter the qualified package name for the report in the format <schema>.<package>. The package name is displayed on the Develop page for the report.	
5. Click Apply or OK to complete the registration.	

The report appears in the resource list of the administration page.

Adding a Custom Report to a Role

To add a custom report to a Warehouse Builder Browser role:

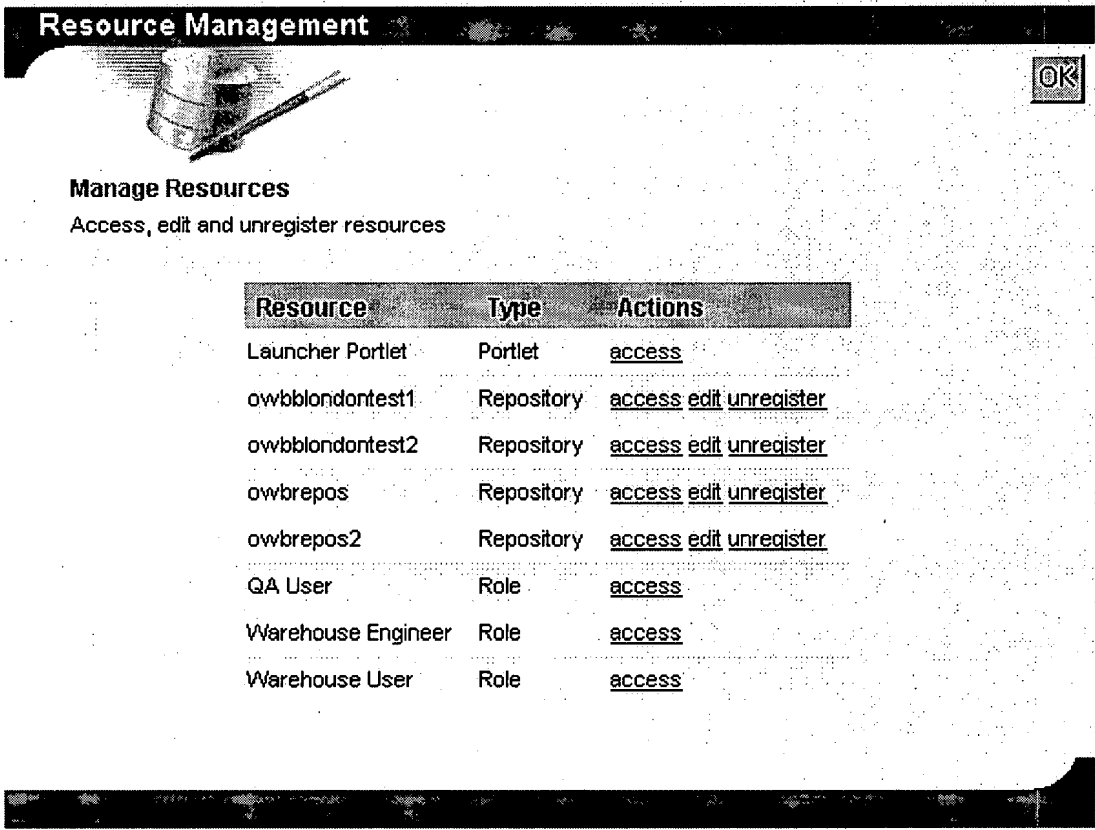
- 1. Click the **roles** action link from the resource list entry for the custom report.
- 2. Click the **add** action link for each role that you want to access the report.

Resource Management

Resource Management allows you to access rights to many aspects of the Warehouse Builder Browser as well as editing, registering, and unregistering repositories and reports.

The Resource Management page contains a table listing all resources that have been registered in the Warehouse Builder Browser. The following sections describe how to modify these resources.

Figure 17-15 Resource Management



Text description of the illustration rsrc mn.gif

Adding User Accounts

To add a user account for Warehouse Builder Browser:

1. From the Resource Management page, select the **Access** action listed next to the resource entitled Launcher Portlet, as listed first in Figure 17-15.

Warehouse Builder Browser displays the Portlet Access page.

Figure 17-16 Portlet Access

Portlet Access : Warehouse Builder Browser

Close

Grant Access to User
 To grant access to a user, select the name of the user and click Grant User.

Grant Access to Group
 To grant access to a group, select the name of the group and click Grant Group.

Change Access
 Click Revoke to deny access for a user or group.

Name	Type	Actions
PORTAL_DEVELOPERS	Group	revoke

Text description of the illustration port_acc.gif

From the Portlet Access page, you can perform the following tasks:

- Grant an access right to a Single Sign On user by selecting the name of the user from the drop-down list and clicking **Grant User**.
- Grant an access right to an Oracle Portal group by selecting the name of the group from the drop-down list and clicking **Grant Group**.
- Revoke an access right by clicking **revoke** in the appropriate table row.

To use the Warehouse Builder Browser, your single sign on user or a group associated with that user must have rights to one of these Warehouse Builder pre-defined roles:

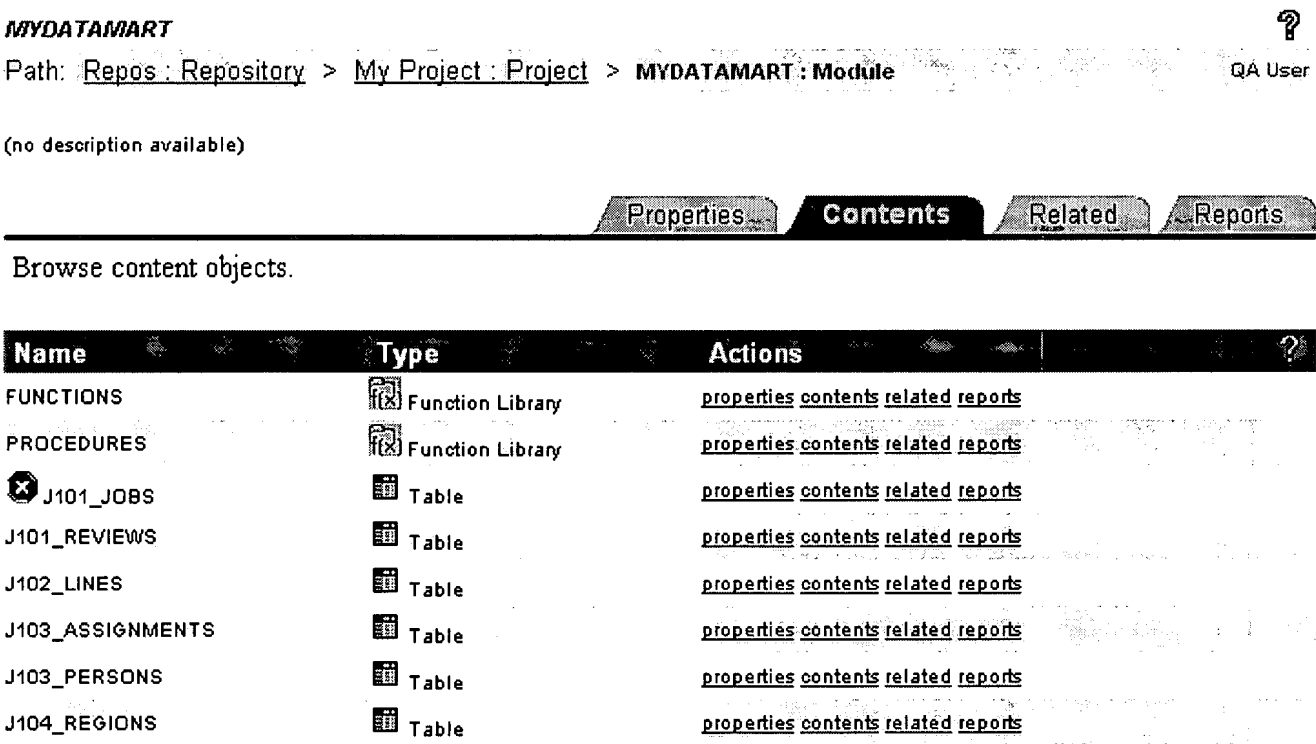
- 3)
- **Warehouse Developer:** A user who uses Warehouse Builder to create the warehouse.
 - **QA User:** A user who tests the quality of the warehouse before it is deployed.
 - **Warehouse User:** A user who uses the deployed warehouse to understand the underlying metadata.

The Administrator can assign these roles to users and groups. All of the pre-defined Reports and

Navigation pages are available to all roles. When you add custom reports, you can assign them to different roles.

For the QA User role, objects that fail validation display an error icon in the Contents tab of the Navigation page. This error icon is not displayed in reports for the other roles.

Figure 17-17 Validation Error



Text description of the illustration qavalida.gif

Managing Custom Reports

The Resource Management page lists all registered custom reports. The actions listed next to the repository are described in Table 17-4.

Table 17-4 Custom Report Management

Actions	Description
Role	Use this to assign a report to one or more roles. When the report is assigned to a role, it appears in the appropriate report list for that role.
Edit	Use this to edit custom report properties.

Actions Description

Unregister Use this to unregister the report. After unregistering the report, it can no longer be browsed using the browser system.

Assigning a custom report to a role adds the report to the appropriate report list page for that role. The name and subject type of the report are indicated at the top left corner of the page. A list of the available roles is provided in the table at the bottom of the page.

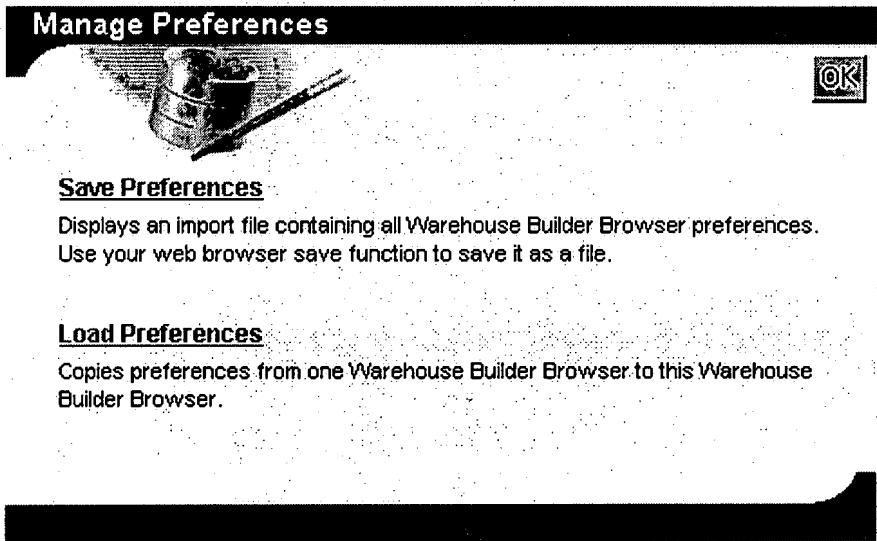
Managing Preferences

Use the Manage Preferences page to save and load Warehouse Builder Browser preferences. This lets you retain your preferences when you upgrade to a new version of Warehouse Builder Browser. You can copy schema preferences across schemas.

The preferences you can save include:

- Favorites.
- Registered custom reports.
- Registered Warehouse Builder Repositories.
- 3-9 • Access rights associated with roles, repositories, custom reports, and the Launcher Portlet.
- External Links.

Figure 17-18 Manage Preferences Page



Text description of the illustration mng_pref.gif

Saving Preferences

To save preferences to a file:

1. Select **Save Preferences** from the Manage Preferences page.

The preferences display in a separate window in text format

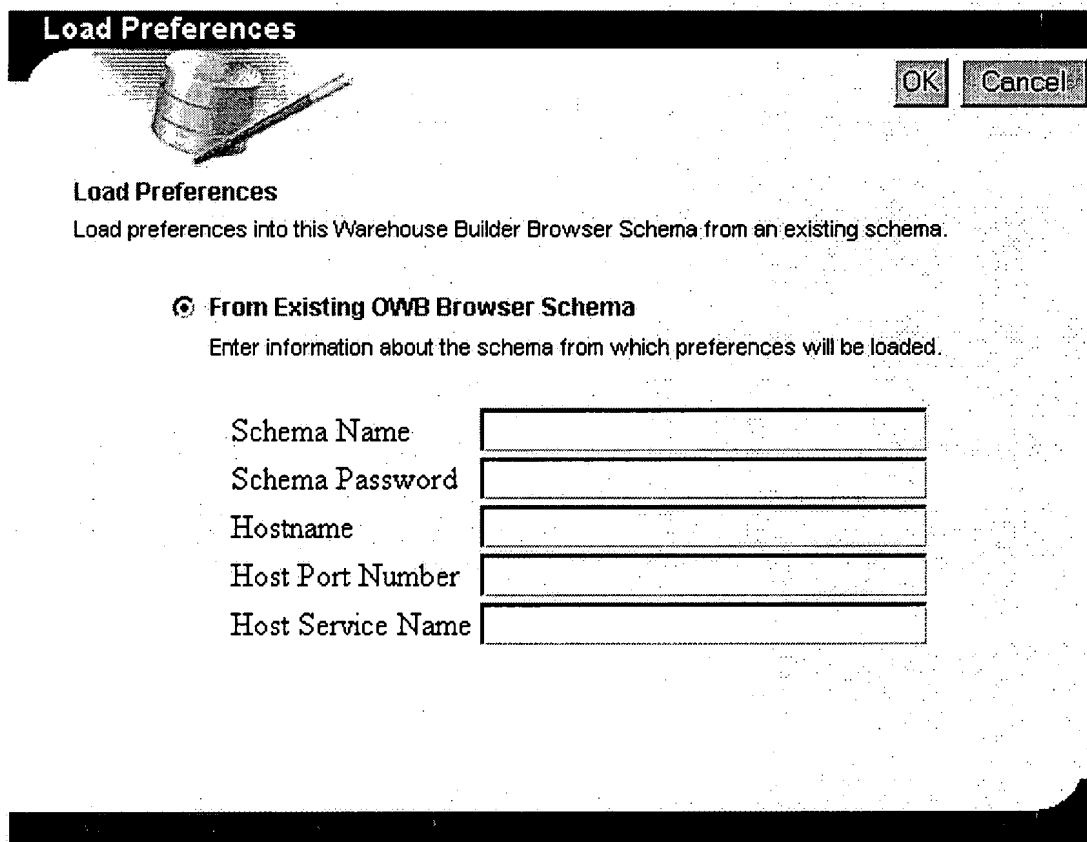
2. From the browser menu bar, select **File**, and then **Save As** to save the file.

This file can be loaded into Warehouse Builder Browser using a tool such as SQL* Plus.

Loading Preferences**To load preferences from an existing schema:**

1. Select **Load Preferences** from the Manage Preferences page.

Figure 17-19 Load Preferences Page



Load Preferences

Load preferences into this Warehouse Builder Browser Schema from an existing schema.

☒ **From Existing OWB Browser Schema**

Enter information about the schema from which preferences will be loaded.

Schema Name	<input type="text"/>
Schema Password	<input type="text"/>
Hostname	<input type="text"/>
Host Port Number	<input type="text"/>
Host Service Name	<input type="text"/>

Text description of the illustration load_pre.gif

2. Specify the following information from an existing Warehouse Builder Browser schema:
 - Schema Name
 - Schema Password

- Hostname
 - Host Port Number
 - Host Service Name
3. Click **OK** to load the preferences into the current Warehouse Builder schema.

A status page displays the preferences that were loaded and any errors that occurred. All errors must be resolved to load the preferences. Errors due to missing database links provide links to the Create Database Links page.

Note:

Database links are not automatically created. If the preferences you are loading contain references to repositories, the database links to those repositories must be created before the load can be successful.

Managing the Dependency Index

Use the Manage Dependency Index page to specify the refresh frequency options for the dependency index for each repository. The dependency index is used to increase performance when running lineage and impact analysis diagrams. You can refresh the dependency index at any time from the Repository page of the Warehouse Builder Navigator.

Setting the Refresh Options

- To specify the dependency refresh option:
1. Open the Warehouse Builder Browser, and select **Manage Dependency Index** from the Administration page or portlet.
- The Manage Dependency Index page displays the available repositories and refresh options.

Figure 17-20 Setting Refresh Options

Repository Name	Refresh Option
owbrepos	Refresh on demand
owbrepos2	Refresh on demand
owbblondontest1	Refresh on demand
owbblondontest2	Refresh on demand

Text description of the illustration mng_di.gif

2. Choose one of the options from the drop-down list and click **OK**. Table 17-5 describes each

option.

Table 17-5 Dependency Index Options

Option	Description
Refresh on demand	<p>You must activate the refresh dependency index link to refresh the index. This link is located on Navigator page that lists all accessible repositories. The dependency index is only refreshed when this action link is activated.</p> <p>This is the best option when using a repository that changes infrequently.</p>
Refresh on first diagram request of the session	<p>Refreshes the dependency index when the first Lineage or Impact Analysis diagram for a repository is run during a session.</p> <p>This is the best option if you want current information, but are not concerned with repository updates that occur during the session.</p>
Refresh on every diagram request	<p>Refreshes the dependency index every time a Lineage or Impact Analysis diagram is requested.</p> <p>This is the best option if you want to display your diagrams and reports with the latest information in the repository.</p>

Refreshing the Dependency Index on Demand

You can refresh the dependency index at any time. If you run a Lineage or Impact Analysis diagram that has never been refreshed, an automatic refresh occurs prior to displaying the diagram.

To refresh the dependency index:

1. Open the Warehouse Builder Browser from the Launcher portlet.

The Contents tab that displays lists the available repositories.

Figure 17-21 Refreshing the Dependency Index

Properties

Contents

Related

Reports

Links

Browse content objects.

Physical Name	Type	Actions
owbblondontest1	Repository	properties contents related reports links refresh dependency index
owbblondontest2	Repository	properties contents related reports links refresh dependency index
owbrepos	Repository	properties contents related reports links refresh dependency index
owbrepos2	Repository	properties contents related reports links refresh dependency index

Text description of the illustration rfrsh_di.gif

2. Select **refresh dependency index**.

The Refresh Dependency Index page displays with a log of previous refreshes at the bottom of the page. The elapsed time helps you determine how long the operation will take.

Figure 17-22 Dependency Index Refresh Log

Refresh Dependency Index

Dependency Index

The Dependency Index is used when displaying Lineage and Impact Analysis diagrams. To receive the most up-to-date information on the Lineage and Impact Analysis diagrams, click on Refresh Dependency Index.

[Refresh Dependency Index](#) [Purge Log](#)

User	Date	Elapsed Time
DEMO	04-OCT-01	.63 seconds
DEMO	04-OCT-01	.7 seconds
DEMO	04-OCT-01	2.42 seconds

Text description of the illustration rfrsh_da.gif

3. Select **Refresh Dependency Index** to refresh the dependency index based on the latest data in the repository.

After the refresh is complete, the log displays the user name, date, and elapsed time of the refresh. You can purge the log by selecting **Purge Log**. This purges the log of the refreshes except for the last refresh.

Other Administration Tasks

To refresh the Portlet Repository:

1. From the Oracle Portal Home Page, click the **Administer** tab.
2. Scroll down to locate the Portlet named Portlet Repository.
3. Click **Refresh Portlet Repository**.
4. Click **Refresh**.

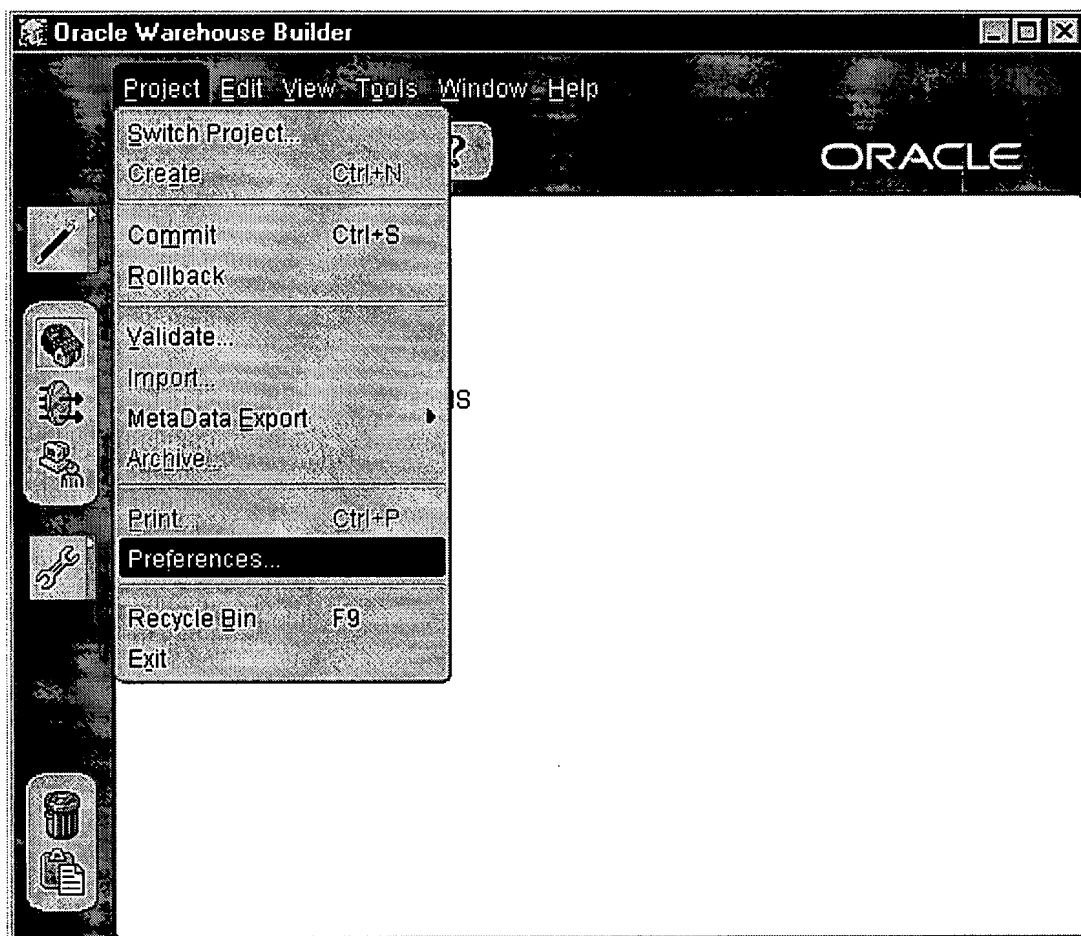
Configuring the Warehouse Builder Client

The Warehouse Builder Console includes a Preferences dialog containing tabs that you use to configure the Warehouse Builder environment. Use the Browser tab to set the network and IP connection information for Oracle Portal. This enables you to view metadata reports using the Warehouse Builder Browser.

To access the preferences dialog:

1. From the **Project** menu, click **Preferences**.

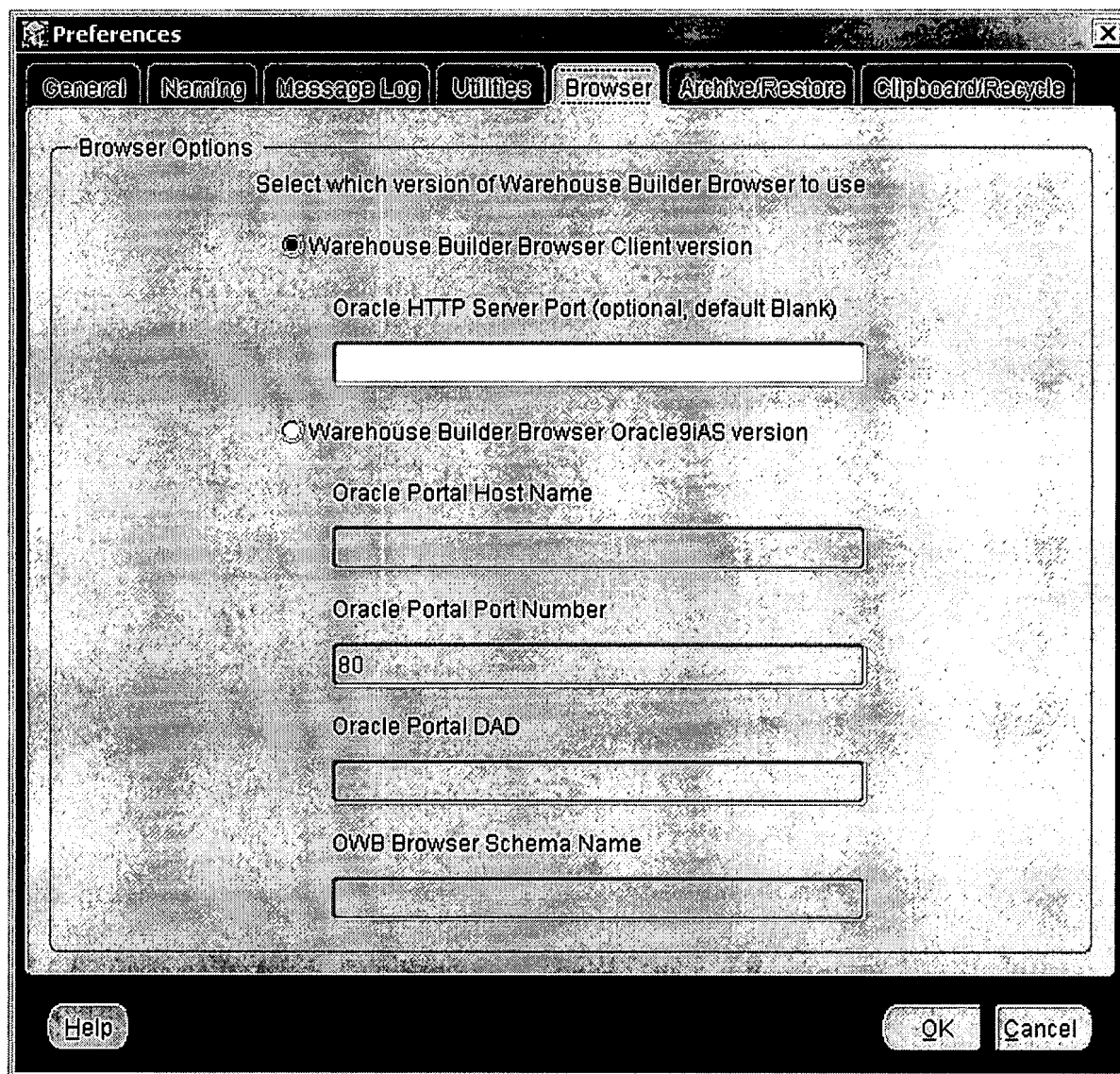
Figure 17-23 Selecting Preferences from the Project Menu



Text description of the illustration preferen.gif

2. From the **Preferences** dialog, select the **Browser** tab.

Figure 17-24 Browser Tab



Text description of the illustration 14 21.gif

3. Specify the following information:
 - Oracle Portal Host Name
 - Oracle Portal Port Number
 - Oracle Portal DAD
 - Warehouse Builder Browser Schema Name
4. Click **OK**.

Creating Custom Reports

You can create custom reports on your metadata using the Warehouse Builder Public Views and Standard Query Language (SQL). These views provide access to your metadata repository tables and report on your data definitions, transformations, and deployment areas. You can use the Warehouse Builder Browser or another reporting tool to view the public views.

For a complete list of available Public Views and the objects they contain, see [Appendix D, "Warehouse Builder Public Views"](#).

Creating a Custom Report in Oracle Portal

To create a custom report using Oracle Portal:

1. Log on to Oracle Portal and select the Database Objects tab from the Navigator page.
2. Find the schema where the Warehouse Builder Browser is loaded, and edit the schema details to ensure that the Application Schema check box is selected.
3. Select the Applications tab, and click the **Create New Application** link.
4. Create a new application and click **OK**.
5. Select the application you just created and select the **Create New Report** link.
6. Select the **Report from SQL Query** link from the page that displays next.
7. Type the report name and display name and click **Next**.
8. Type the SQL query to define the report and click **Finish**.

Click **Next** to continue to pages where you can customize the appearance of the report. To customize the report at a later time, select the Edit action.

Your SQL queries must reference a database link to the Warehouse Builder repository. You can use the default_owb_link created during the Warehouse Builder Browser installation. The SQL query for a report can only call PUBLIC database link or links within the application schema where a report resides.

Although a reports can reside in a schema other than Warehouse Builder Browser schema, the report must be executable by the Warehouse Builder Browser schema. To grant execution privilege for a portal report, go to **Oracle Portal Home Page > Database Objects > Database Schemas > Report Schema > Report Package > Grant Access**.

The following query provides a simple project report that lists the information systems it contains:

```
select * from all_iv_information_systems@default_owb_link where project_id = :id
```

When run from the Warehouse Builder Browser Navigation pages, the marker :id is automatically substituted with the appropriate value.

Verify that the report can be run in the following environments:

- **SQL*Plus:** Log on as the user who owns the report. The owner is displayed on the Develop page for the report. In SQL* Plus, replace the marker :id with a valid project_id.
- **Oracle Portal:** From the Develop page, select the **Customize** link, enter a valid project_id in the edit box labelled Id, and click **Run Report**.

		<u>Oracle</u>	<u>Go To Table</u>	<u>Go To</u>
<u>Go to previous</u>	<u>Go to next</u>	<u>Copyright © 1996, 2003 Oracle Corporation.</u>	<u>Of Contents</u>	<u>Index</u>
<u>page</u>	<u>page</u>	All Rights Reserved.	<u>Contents</u>	<u>Index</u>

[Skip Headers](#)

Oracle9i Warehouse Builder User's Guide
Release 2 (9.0.4)
Part Number B10657-01

Go To Table	Go To
Of Contents	Index
Contents	Index

[Go to previous page](#) [Go to next page](#)

18

Extending Warehouse Builder Functionality

This chapter describes how to extend current Warehouse Builder functionality. This chapter includes the following topics:

- [About Oracle Metabase \(OMB\) Plus](#)
- [User-Defined Properties](#)
- [Managing Security with PL/SQL](#)

About Oracle Metabase (OMB) Plus

This chapter discusses tasks you can perform to extend Warehouse Builder functionality via Oracle Metabase (OMB) Plus.

OMB Plus is the flexible, high-level command line metadata access tool for Oracle9i Warehouse Builder. With OMB Plus, you can write the syntactic constructs such as variable support, conditional and looping control structures, error handling, and standard library procedures. You can access the Warehouse Builder metadata repository and the runtime repository. Using OMB Plus, you can navigate repositories and manage and manipulate metadata in repositories.

The remainder of this chapter discusses how to perform specialized tasks in Warehouse Builder using the OMB scripting language. For syntax information on specific OMB Plus commands, see *Oracle9i Warehouse Builder Scripting Reference*.

User-Defined Properties

Warehouse Builder allows you to extend its design repository through User Defined Properties (UDP). Each repository object has a pre-defined property set. You can add custom properties to an object by creating a UDP.

- 1) You create and manage UDPs using the Oracle MetaBase (OMB) Scripting Language. You can view UDPs using OMB or the Warehouse Builder client. In the client, UDPs display on property sheets and in the Warehouse Builder Browser.

UDPs behave like native properties and follow Warehouse Builder rules for object locking, multiuser access, transactions, and security. When you take metadata snapshots of the object, Warehouse Builder captures the associated UDPs. You can also import and export UDPs using the Metadata Loader (MDL).

Managing User Defined Properties

As the Warehouse Builder administrator, you should define all user-defined properties into the Warehouse Builder repository before allowing end users to access it. In doing so, you avoid the task of supplying values for UDPs on existing objects.

You should register all user-defined properties centrally in the design repository, and not locally on the client. To create or edit a UDP, you must be the single user accessing to the Warehouse Builder repository.

You can use the following OMB Plus commands for creating and manipulating user defined properties:

- OMBDEFINE
- OMBDESCRIBE

When you create and commit a UDP, OMB performs the following validations:

- A namespace check ensures that you did not define two identically named properties within the same class hierarchy. Prefix the user defined property with 'UDP_' to avoid conflicts with the future names introduced by Warehouse Builder.
- A property value check ensures that you defined default values consistent with the data types you specified.
- A user access check ensures that you have single-user access to the entire repository.

OMBDEFINE

The OMBREDEFINE CLASS_DEFINITION allows you to manipulate UDPs. To create a UDP on the Dimension object, issue the following statement. This adds a UDP definition to class definition 'DIMENSION':

```
OMBREDEFINE CLASS_DEFINITION 'DIMENSION'
  ADD PROPERTY_DEFINITION 'UDP_Dim' SET PROPERTIES (TYPE, DEFAULT_VALUE)
  VALUES ('INTEGER', '100')
```

The following command adds a property to the 'COLUMN' type. This property displays in the Table, View, Materialized View, External Table and Sequence Property Sheets:

```
OMBREDEFINE CLASS_DEFINITION 'COLUMN'
  ADD PROPERTY_DEFINITION 'UDP_Col' SET PROPERTIES (TYPE, DEFAULT_VALUE)
  VALUES ('STRING', 'foo')
```

The following command allows you to change the name or the default value of a given property.

```
OMBREDEFINE CLASS_DEFINITION 'TABLE' MODIFY PROPERTY_DEFINITION 'UDP_Tbl'
```

```
SET PROPERTIES (DEFAULT_VALUE, BUSINESS_NAME)
VALUES ('99', 'UDP_Tbl')
```

The following command deletes the `tbl_udp` property from the 'Table' class. This is a very destructive and highly deprecated action since it cannot be undone. It renders all property value customizations made for this property definition in your repository irretrievable:

```
OMBREDEFINE CLASS_DEFINITION 'TABLE' DELETE PROPERTY_DEFINITION 'UDP_Tbl'
```

OMBDESCRIBE

You can use `OMBDESCRIBE` on a Class Definition to view the attributes for a metadata element. `OMBDESCRIBE` allows you to list the user defined properties you for a given object type. For instance, the following command lists the user defined properties for a dimension:

```
OMBDESCRIBE CLASS_DEFINITION 'DIMENSION' GET PROPERTY_DEFINITIONS
```

You can also use `OMBDESCRIBE` to introspect the properties of a Property Definition. For instance, for a user defined property called `UDP_Dim` under the Dimension Class Definition, you can learn the datatype, default value and business name with the following command:

```
OMBDESCRIBE CLASS_DEFINITION 'DIMENSION' PROPERTY_DEFINITION 'UDP_Dim'
GET PROPERTIES (TYPE, DEFAULT_VALUE, BUSINESS_NAME)
```

You can specify the data type of a user-defined property such as char, number and date.

Viewing User Defined Properties

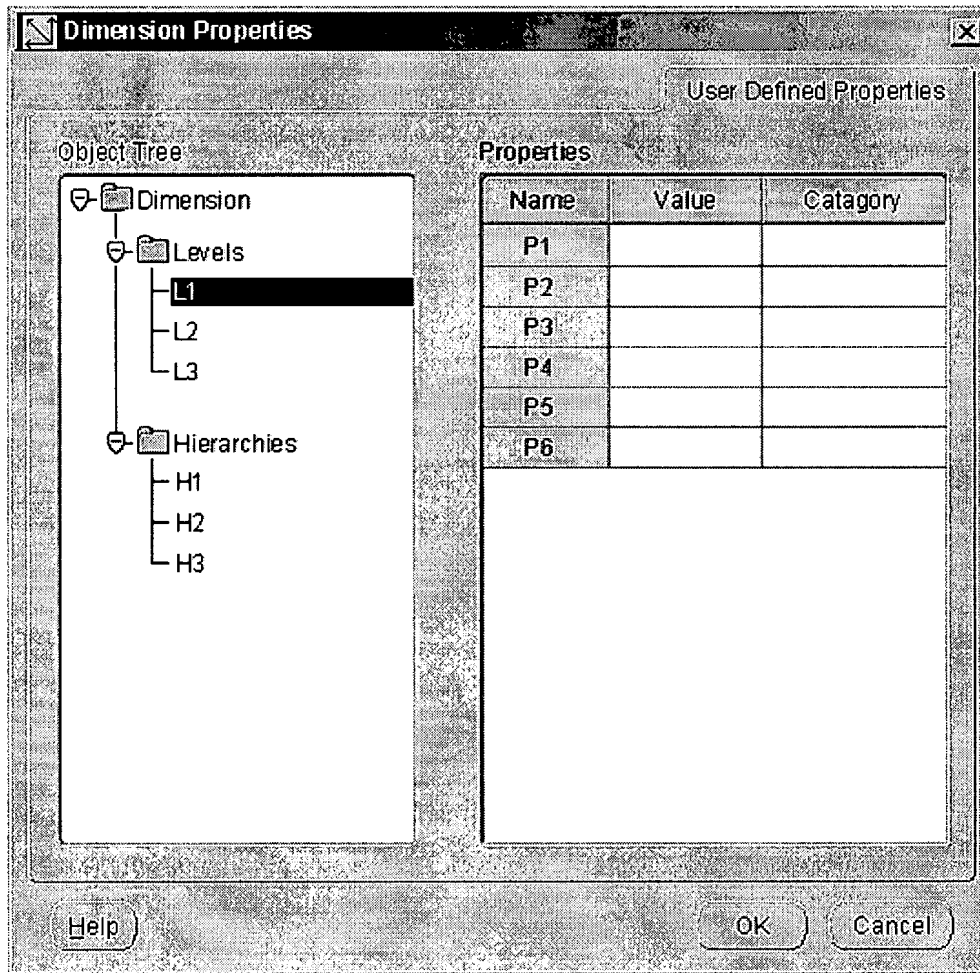
In the user interface, you can view UDPs in the following components:

- [Warehouse Builder Client](#)
- [Warehouse Builder Design Browser](#)

Warehouse Builder Client

Once you create a UDP using scripting, Warehouse Builder displays the UDP in the User Defined Properties tab on the associated properties sheet. The User Defined Properties tab does not appear until you create a UDP. For example, the Dimension Properties sheet typically does not display the User Defined Properties tab. However, once you add a UDP to a dimension, the UDP appears on the Dimension Properties sheet shown in [Figure 18-1](#).

Figure 18-1 Sample Properties Sheet with User-Defined Properties



Text description of the illustration samplepr.gif

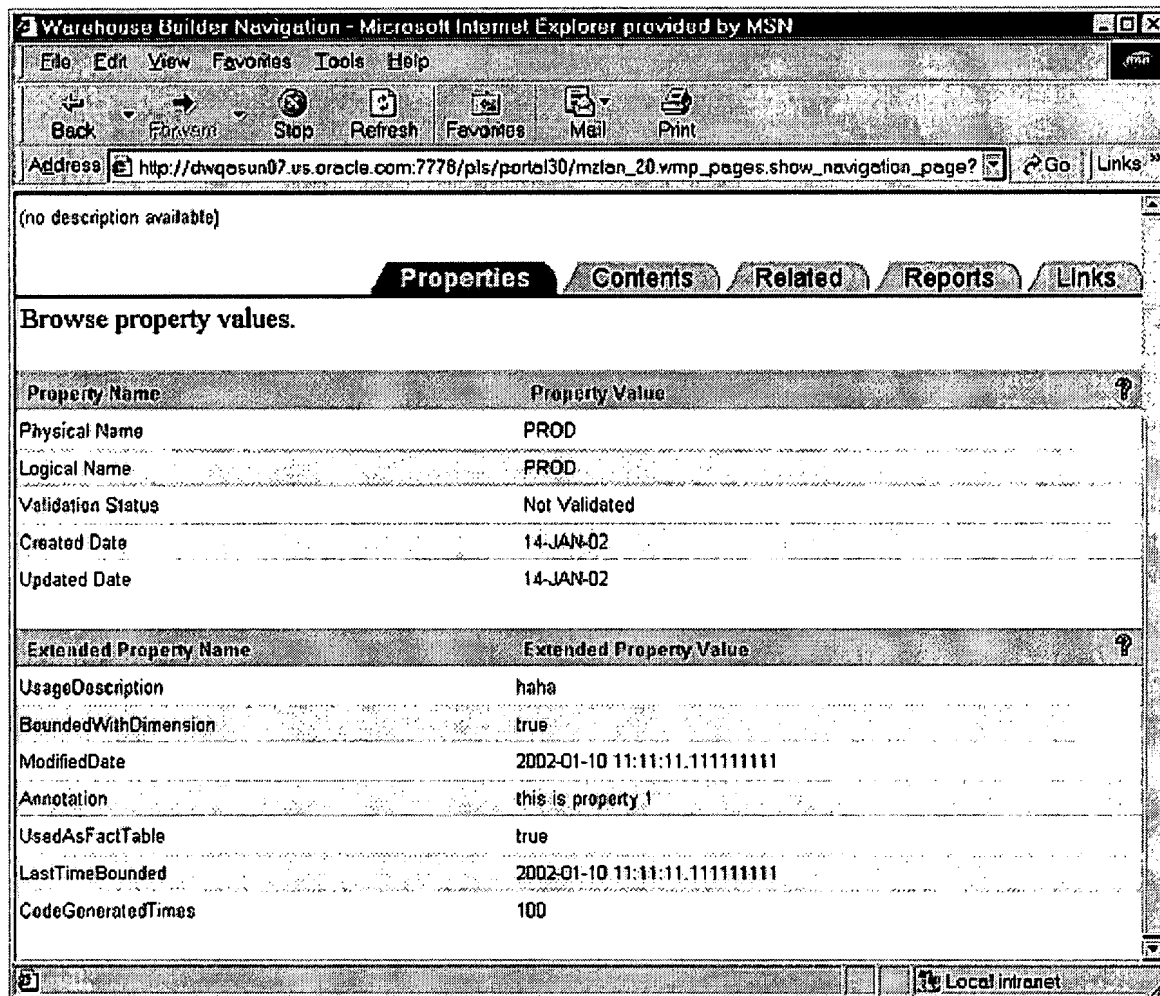
In the left panel, the tab displays the object navigation tree. In the right panel, the tab lists all the contained objects and corresponding extended properties. It shows the values and categories you specified when you created the UDP in scripting. You can modify the values but not the categories in the User Defined Properties tab. You must use OMB Plus to edit categories.

Warehouse Builder Design Browser

The Warehouse Builder Browser also displays UDPs. Warehouse Builder Browser is a metadata management and reporting portal for Warehouse Builder. Browser displays object properties, object relationships, and lineage and impact analysis reports.

If you define a UDP for a given object, the Browser lists the UDP name and values as Extended Property Name and Extended Property Value as shown in Figure 18-2.

Figure 18-2 Sample Properties Sheet with User-Defined Properties



Text description of the illustration [samplebr.gif](#)

Transferring UPDs to Other Repositories

2) The primary method for propagating changes from one repository to another is via MDL. The MDL allows you to export and import the metadata definition of the user-defined properties and its contents.

Exporting UDPs

You can export UDPs from the command line only. In the MDL Control file, the option is `DEFINITIONFILE=filename` to export the metadata definition. For example:

```
## Sample Export file
USERID=UserName/Password@HostName:PortID:OracleServiceName
#
DEFINITIONFILE=Drive:\DirectoryName\filename.mdd

FILE=Drive:\DirectoryName\filename.mdl
LOG=Drive:\DirectoryName\filename.log
```

Importing UDPs

You can import UDPs from the command line only. During import, MDL updates the user-defined

properties for all objects. In the MDL Control file, the option is `DEFINITIONFILE=filename` to import the metadata definition. For example:

```
## Sample Import file
USERID=UserName/Password@HostName:PortID:OracleServiceName
#
DEFINITIONFILE=Drive:\DirectoryName\filename.mdd

FILE=Drive:\DirectoryName\filename.mdl
LOG=Drive:\DirectoryName\filename.log
```

You can import UDPs using one of the following search criteria:

- **Universal ID:** The metadata definition contains a Universal Object ID (UUID). The UUID uniquely identifies objects across repositories. If you import the MDL file by UUID, then MDL looks up the metadata definition by UUID. If the metadata definition name in the source MDL file is different from the metadata definition in the repository, then MDL renames it.
- **Physical Name:** MDL looks up the metadata definition by physical name.

Regardless of the import mode, MDL either adds the metadata definition if it does not exist in the repository, or updates the metadata definition if it already exists. MDL will not delete metadata definitions in the repository.

When updating the metadata definition, MDL only renames the object if the names are different (search criteria is by UUID), and update the default value. MDL does not change the datatype (e.g., STRING).

Managing Security with PL/SQL

This section contains the following topics:

- Maintaining Repository Users
- Plug-in Interface for the Security PL/SQL Package Specification
- Definitions of the Constants in the Package Specification
- Implementing the PL/SQL Interface

Maintaining Repository Users

3) Multiple identifiable Warehouse Builder users can access the same central repository schema when they are registered by the repository owner.

Warehouse Builder includes utility procedures for the following maintenance tasks:

- Update the role protecting password

Although the repository owner does not explicitly use the protecting password for the role, it is recommended that the repository owner change the password often.

The repository owner must connect to the repository schema using SQL Plus and issue the following statement:

```
Call WBSecurityHelper.updateRolePwd('newpwd');
```

where 'newpwd' is the new password used to protect the role chosen by repository owner. The changed password encryption appears in the table called OWB_ROLE_INFO table.

3)

- Register repository users

To register a repository user, execute the following statement in SQL Plus:

```
call WBSecurityHelper.registerOWBUser ('username')
```

- Unregister repository users

To unregister a user from a repository, execute the following statement in SQL Plus:

```
Call WBSecurityHelper.unregisterOWBUser('username');
```

- List all repository users

The repository owner must connect to the database containing the repository and issue the following statement:

```
Set serveroutput on;
Call WBSecurityHelper.listOWBUser( );
```

Set serveroutput on is required since listOWBUser() uses the DBMS_OUTPUT.put_line for the output to dump the data to the user interface. Otherwise, DBMS_OUTPUT.put_line only dumps the output into an intermediate data structure.

Plug-in Interface for the Security PL/SQL Package Specification

This section describes the plug in interface specification for the PL/SQL security package provided by Warehouse Builder. You need to substitute the dummy PL/SQL package body provided by Warehouse Builder by implementing this interface in your Warehouse Builder repository. The interface specification and the dummy implementation are also available from your repository schema.

```
CREATE OR REPLACE PACKAGE WBSecurityServiceImpl AS

FUNCTION isSecurityServiceCustomized RETURN NUMBER;

/*
```

Use Function: isSecurityServiceCustomized() to differentiate the implementation of security service, whether you customize the security PL/SQL implementation or use the implementation provided by Warehouse Builder.

Return value: this function returns 1 if you want to implement the PL/SQL package of this specification,

otherwise it returns 0;

*/

```
PROCEDURE securityCheckForCreation(outcome OUT NUMBER,
  userId IN VARCHAR2,
  objectUUIDOperationInvokedOn IN VARCHAR2,
  status IN VARCHAR2,
  parentModuleUUID IN VARCHAR2,
  parentProjUUID IN VARCHAR2,
  repos_Schema IN VARCHAR2,
  objectType IN NUMBER);
```

/*

Procedure: securityCheckForCreation: Used to create an operation security check. When you try to create an object, Warehouse Builder calls this procedure to ask the implementation whether the creation operation is acceptable or not.

Argument explanation:

Outcome: 1: The creation operation is acceptable.

Outcome: 0: The creation operation is not acceptable.

userId: The login user's database user name.

objectUUIDOperationInvokedOn: The parent folder UUID, where you create a new object. For objects, such as Projects and Snapshots, that do not have folder parent, this argument is NULL.

status: An attribute of a module: WB_DEV_STATUS, WB_QA_STATUS, WB_PROD_STATUS defined in this specification. This attribute describes the status of the module. For objects, such as Projects, Modules, or Snapshots, that are not children of any module in the hierarchy, this argument is NULL.

parentModuleUUID: The UUID of the module. For objects such as Projects or Snapshots that are not children of any module in the hierarchy, this argument is NULL.

parentProjUUID: The UUID of the project. For objects such as Projects or Snapshots that are not children of any module in the hierarchy, this argument is NULL.

repos_schema: The central repository schema name you are working on.

objectType: The type of object you want to create. It is one of the object type constants defined in this specification.

*/

```
PROCEDURE securityCheck(outcome OUT NUMBER,
  userId IN VARCHAR2,
  operation IN NUMBER,
  objectUUIDOperationInvokedOn IN VARCHAR2,
  objectTypeOperationInvokedOn IN NUMBER,
  status IN VARCHAR2,
  parentModuleUUID IN VARCHAR2,
```

```
parentProjUUID    IN VARCHAR2,
repos_Schema IN VARCHAR2);
```

```
/*
```

PROCEDURE: securityCheck is used for the following operations:

```
WB_EDIT
WB_DELETE
WB_VALIDATE
WB_GENERATION
WB_VERSION
```

Whenever you invoke one of the above listed operations, Warehouse Builder calls this procedure to see whether the operation is acceptable or not.

Argument explanation:

Outcome: 1: The creation operation is acceptable.

Outcome: 0: The creation operation is not acceptable.

userId: The login user's database user name.

operation: One of constants listed above.

objectUUIDOperationInvokedOn: The target object's UUID.

objectTypeOperationInvokedOn: The type of object the operation is invoked on (one of the object type constants defined in this specification).

status: An attribute of a module.

```
WB_DEV_STATUS
```

```
WB_QA_STATUS,
```

```
WB_PROD_STATUS
```

This attribute is used to describe the status of the module. If you operate on a project, module, or snapshot, this argument is NULL.

parentModuleUUID: The UUID of the module. If you invoke the operation on a module, then the objectUUIDOperationInvokedOn and the parentModuleUUID are the same. If you invoke an operation on a project or snapshot that is not a child of any module in the hierarchy, this argument is NULL.

parentProjUUID: The UUID of the project. If you invoke the operation on a project, then the objectUUIDOperationInvokedOn and the parentProjUUID are the same. If you invoke the operation on a snapshot that is not the child of any project in the hierarchy, this argument is NULL.

repos_schema: The central repository schema name you are working on

```
*/
```

```
PROCEDURE securityCheckForService(outcome OUT NUMBER,  
    userId IN VARCHAR2,  
    serviceOp IN NUMBER,  
    moduleUOID IN VARCHAR2,  
    projUOID IN VARCHAR2,  
    repos_Schema IN VARCHAR2);
```

```
/*
```

PROCEDURE securityCheckForService is used for the following service operations:

WB_DEPLOY

WB_MDL_IMPORT

WB_MDL_EXPORT

WB_BRIDGE_IMPORT

WB_BRIDGE_EXPORT

WB_SOURCE_IMPORT

WB_RUNTIME_EXECUTE

WB_SNAPSHOT_RESTORE

Argument explanation:

Outcome: 1: The creation operation is acceptable.

Outcome: 0: The creation operation is not acceptable.

userId: The login user's database user name.

serviceOp: One of constants listed above.

moduleUOID: The UOID of a module on which the user is invoking operation:serviceOp. The result is NULL for serviceOp:WB_DEPLOY,WB_BRIDGE_EXPORT, WB_RUNTIME_EXECUTE. It is valid for: WB_MDL_IMPORT, WB_MDL_EXPORT, WB_SOURCE_IMPORT, WB_SNAPSHOT_RESTORE.

projUOID: The UOID of a project on which the user is invoking the operation: serviceOp on it. The result is NULL for the serviceOp:WB_DEPLOY, WB_BRIDGE_EXPORT, WB_RUNTIME_EXECUTE. It is valid for: WB_MDL_IMPORT, WB_MDL_EXPORT, WB_SOURCE_IMPORT, WB_SNAPSHOT_RESTORE.

repos_schema: The central repository schema name you are working on

```
*/
```

--BEGIN CONSTANT DEFINITION

```
CUSTOM_SHARED_LIBRARY CONSTANT
VARCHAR2(100):='9E012195D16211D48D7100B0D02A59E8';
```

```
/*
```

CUSTOM_SHARED_LIBRARY: The UUID constant for a predefined Warehouse Builder folder: Custom shared library. On the UI, this folder is called Custom located under the Public Transformation node.

This folder contains all global shared library transformations created by the user.

To control the users who have permission to create transformations under the Custom folder, you can check if the argument object `UUIDOperationInvokedOn` in procedure `securityCheckForCreation` equals to this constant or not. To control users who can invoke the operations such as `WB_EDIT` or `WB_DELETE`, on a shared function or a procedure, check if the argument `parentModuleUUID` in the procedure `securityCheck` equals this constant or not.

Since this module is predefined, you cannot change the status or use the status of this module for access control.

```
*/
```

--Definition of constants for all basic operations

```
WB_EDIT    CONSTANT INTEGER := 0;
WB_DELETE  CONSTANT INTEGER := 1;
WB_REFERENCE CONSTANT INTEGER := 2;
WB_CREATE  CONSTANT INTEGER := 3;
WB_VALIDATE CONSTANT INTEGER := 4;
WB_GENERATION CONSTANT INTEGER := 5;
WB_VERSION CONSTANT INTEGER := 6;
```

```
/*
```

Use this group of operation constants in the `SecurityCheck` procedure. Use one of the above constants for the argument "operation". You can also use it to control which user can invoke this operation.

```
*/
```

--Definition of constants for all service type operations

```
WB_DEPLOY    CONSTANT INTEGER := 100;
WB_MDL_IMPORT CONSTANT INTEGER := 101;
WB_MDL_EXPORT CONSTANT INTEGER := 102;
WB_BRIDGE_IMPORT CONSTANT INTEGER := 103;
WB_BRIDGE_EXPORT CONSTANT INTEGER := 104;
WB_SOURCE_IMPORT CONSTANT INTEGER := 105;
WB_RUNTIME_EXECUTE CONSTANT INTEGER := 106;
WB_SNAPSHOT_RESTORE CONSTANT INTEGER := 107;
```

```
/*
```

Use this group of service operation constants in `SecurityCheckForService` procedure. Use one of these constants for the argument "serviceOp". You can also use it to control which user can invoke this service operation.

*/

--Definition of the module status

```
WB_DEV_STATUS  CONSTANT VARCHAR2(100) := 'DEV_STATUS';
WB_QA_STATUS   CONSTANT VARCHAR2(100) := 'QA_STATUS';
WB_PROD_STATUS CONSTANT VARCHAR2(100) := 'PROD_STATUS';
```

/*

Use this group of module status constants should be used in the securityCheckForCreation and SecurityCheck procedures. Use the above constants if you want to create a child of a module (for securityCheckForCreation) or if you invoke an operation on a module or the child of a module (for SecurityCheck), otherwise the argument "status" will be null. You can also use it to implement your access control based on the status of the module.

*/

--Definition of object type

```
WB_PROJECT      CONSTANT INTEGER := 1;
WB_ORACLE_MODULE CONSTANT INTEGER := 2;
WB_GATEWAY_MODULE CONSTANT INTEGER := 3;
WB_SAP_MODULE   CONSTANT INTEGER := 4;
WB_FLAT_FILE_MODULE CONSTANT INTEGER := 5;
WB_SHARED_MODULE CONSTANT INTEGER := 6;
WB_REPOS_MODULE CONSTANT INTEGER := 7;
WB_COLLECTION   CONSTANT INTEGER := 8;
WB_WAREHOUSE    CONSTANT INTEGER := 9;
WB_TABLE        CONSTANT INTEGER := 10;
WB_VIEW         CONSTANT INTEGER := 11;
WB_MATERIALIZED_VIEW CONSTANT INTEGER := 12;
WB_SEQUENCE     CONSTANT INTEGER := 13;
WB_DIMENSION_TABLE CONSTANT INTEGER := 14;
WB_CUBE_TABLE   CONSTANT INTEGER := 15;
WB_FLAT_FILE    CONSTANT INTEGER := 16;
WB_PACKAGE      CONSTANT INTEGER := 17;
WB_TRANSFORMATION CONSTANT INTEGER := 18;
WB_MAPPING      CONSTANT INTEGER := 19;
WB_MIV_MODULE   CONSTANT INTEGER := 20;
WB_CONNECTOR    CONSTANT INTEGER := 21;
WB_LOCATION     CONSTANT INTEGER := 22;
WB_RUNTIME_REPOSITORY CONSTANT INTEGER := 23;
WB_BUSINESS_AREA CONSTANT INTEGER := 24;
WB_INTELLIGENCE_MODULE CONSTANT INTEGER := 25;
WB_PROCESS_FLOW CONSTANT INTEGER := 26;
WB_PROCESS_FLOW_MODULE CONSTANT INTEGER := 27;
WB_PROCESS_FLOW_PACKAGE CONSTANT INTEGER := 28;
WB_QUERY_OBJECT CONSTANT INTEGER := 29;
WB_ADVANCED_QUEUE CONSTANT INTEGER := 30;
WB_EXTERNAL_TABLE CONSTANT INTEGER := 31;
WB_REPORT        CONSTANT INTEGER := 32;
WB_REPORT_GROUP  CONSTANT INTEGER := 33;
WB_REPORT_MODULE CONSTANT INTEGER := 34;
WB_OBJECT_TYPE   CONSTANT INTEGER := 35;
WB_SNAPSHOT      CONSTANT INTEGER := 36;
```

/*

This group of object type constants must be used in procedures securityCheckForCreation and

securityCheck. The argument "objectType" in securityCheckForCreation and argument "objectTypeOperationInvokedOn" in securityCheck will be one of the above constants. You can use this constant to control which user can create which kind of object or which user can invoke operations such as WB_EDIT or WB_DELETE on what types of objects.

*/

/*

Because there are many arguments, the procedures will depend on whether the object on which the user invokes an operation is the child of the project or a module. The information is listed below.

*/

--Children of Project

/*

While WB_SNAPSHOT and WB_PROJECT are not children of a project, other objects in the list of type constants are.

*/

--Children of Module

/* The following are not children of any module: WB_PROJECT, WB_ORACLE_MODULE, WB_GATEWAY_MODULE, WB_SAP_MODULE, WB_FLAT_FILE_MODULE, WB_SHARED_MODULE, WB_REPOS_MODULE, WB_COLLECTION, WB_WAREHOUSE, WB_MIV_MODULE, WB_LOCATION, WB_CONNECTOR, WB_RUNTIME_REPOSITORY, WB_BUSINESS_AREA, WB_INTELLIGENCE_MODULE, WB_PROCESS_FLOW_MODULE, WB_REPORT_MODULE, WB_SNAPSHOT.

Other object types can be the children of module.

*/

END WBSecurityServiceImpl;

Warning

For the following procedure defined in the above specification:

```
PROCEDURE securityCheckForService (outcome OUT NUMBER,

userID IN VARCHAR2,
serviceOp IN NUMBER,
moduleUOID IN VARCHAR2
projUOID IN VARCHAR2
repos_Schema IN VARCHAR2) ;
```

For this release, do not use the arguments: moduleUOID and projUOID. In later releases Warehouse

Builder will provide module and project level security check for the following service operations: `SNAPSHOT_RESTORE`, `SOURCE_IMPORT`, `MDL_IMPORT`, `MDL_EXPORT`, and `BRIDGE_IMPORT`.

Definitions of the Constants in the Package Specification

The following section lists the constant definitions of basic operations. For these operations, Warehouse Builder checks if the operation is acceptable or not on an object instance level

- **WB_EDIT**: Edit (update) operation
- **WB_DELETE**: Delete operation
- **WB_REFERENCE**: The operation you want to reference an object within another object. For example, when you want to drag a table into a mapping as an operation.
- **WB_CREATE**: Create operation
- **WB_VALIDATE**: Validate operation used to check the accuracy of the object definitions
- **WB_GENERATION**: Generate the SQL scripts for the definitions in Warehouse Builder
- **WB_VERSION**: Versioning objects which means user can add an object to a snapshot version, remove an object from a snapshot version, or replace an existing versioned object with a new copy.

The following section lists the constant definitions for service operations. For these operations, Warehouse Builder checks if the operation is acceptable or not on a system wide level. If a user has the privilege to invoke a service operation, he should invoke this operation on any object. For operations such as `WB_MDL_IMPORT`, `WB_MDL_EXPORT`, `WB_BRIDGE_IMPORT`, `WB_SOURCE_IMPORT`, `WB_SNAPSHOT_RESTORE`, Warehouse Builder provides security at the project or module level.

- **WB_DEPLOY**: First generates the SQL script (if does not exist) and then deploy the SQL scripts to a runtime database schema.
- **WB_MDL_IMPORT**: Imports the metadata into Warehouse Builder from a flat file format.
- **WB_MDL_EXPORT**: Exports the metadata from Warehouse Builder into a flat file.
- **WB_BRIDGE_IMPORT**: Another importing solution.
- **WB_BRIDGE_EXPORT**: Another exporting solution.
- **WB_SOURCE_IMPORT**: Imports the metadata information of database objects, such as tables and views, into Warehouse Builder from a given database link.
- **WB_RUNTIME_EXECUTE**: Executes the deployed SQL script from the runtime database schema.
- **WB_SNAPSHOT_RESTORE**: Restores the design space from a given snapshot, selectively or fully.

Constant definitions of the module status. You can change the module status from the Warehouse Builder client using the property page of a module.

- **WB_DEV_STATUS**: The module is under development.
- **WB_QA_STATUS**: The module is in QA status.
- **WB_PROD_STATUS**: The module is in production.

Constant definitions of object types.

- **WB_PROJECT**
- **WB_ORACLE_MODULE**
- **WB_GATEWAY_MODULE**
- **WB_SAP_MODULE**
- **WB_FLAT_FILE_MODULE**
- **WB_SHARED_MODULE**
- **WB_REPOS_MODULE**
- **WB_COLLECTION**
- **WB_WAREHOUSE**
- **WB_TABLE**
- **WB_VIEW**
- **WB_MATERIALIZED_VIEW**
- **WB_SEQUENCE**
- **WB_DIMENSION_TABLE**
- **WB_CUBE_TABLE**
- **WB_FLAT_FILE**
- **WB_PACKAGE**
- **WB_TRANSFORMATION**
- **WB_MAPPING**
- **WB_MIV_MODULE**
- **WB_CONNECTOR**

- WB_LOCATION
- WB_RUNTIME_REPOSITORY
- WB_BUSINESS_AREA
- WB_INTELLIGENCE_MODULE
- WB_PROCESS_FLOW
- WB_PROCESS_FLOW_MODULE
- WB_PROCESS_FLOW_PACKAGE
- WB_QUERY_OBJECT
- WB_ADVANCED_QUEUE
- WB_EXTERNAL_TABLE
- WB_REPORT
- WB_REPORT_GROUP
- WB_REPORT_MODULE
- WB_OBJECT_TYPE
- WB_SNAPSHOT

Implementing the PL/SQL Interface

When you implement the PL/SQL interface, you decide whether an operation is accepted or rejected based on the arguments passed by the Warehouse Builder client. Warehouse Builder also provides a public view named `ALL_IV_FIRSTCLASS_OBJECTS` in its repository. This view contains the following information about an object:

- Object UUID and class type
- Object name and business name
- Created by and updated by
- Creation and update timestamps

Given the object UUID, you can reference the public view for the above information. You can use this information from the public view to decide whether the operation is acceptable or not.

The arguments passed by the caller provide you with different levels of security granularity:

- **Repository level:** Check the `repos_Schema` argument as to whether to freeze the whole repository.

- **Project level:** Check the argument `parentProjUUID` to freeze the whole project or not.
- **Installed module level:** Check the argument `parentModuleUUID` to freeze the whole installed module or not.
- **Object class type level:** Check the argument `objectTypeOperationInvokedOn` to accept or prevent access to the entire type.
- **Object instance level:** Check the object UUID.
- **Development process level:** Check the argument: `status` (development, quality assurance, or production) to make access control decisions for an entire module. The status is an attribute of any module that you can change using the Warehouse Builder user interface through the module properties page.

Considerations

The following are considerations for implementing the PL/SQL interface:

- If you base your access control decisions on the status of a module, Warehouse Builder recommends that only users in the administration group have the access to edit the module.
- If a user has the validate/generate privilege on an object instance, such as a Table, and has no edit privilege on the object instance, the user can only validate/generate the object without the generated result persisting.
- When you invoke an operation (operation A) from the Warehouse Builder client, in rare cases, operation B is invoked underneath. This can result in a situation where you have granted a user permission to invoke operation A, but the user can not successfully finish operation A because s/he has no permission to invoke operation B.

To resolve this issue, you need to either grant the user privileges for both operations A and B, or not grant the user privileges on either operation.

For example, in the mapping editor, when a bound object is added to a mapping, or when inbound reconcile is performed on an object, Warehouse Builder automatically creates a connector to the bound object's location if one does not already exist. To allow the user to finish adding a bound object into mapping editor, or to finish the inbound reconcile, you need to grant the user `EDIT` permission on the mapping and `CREATE` permission for the connector.

- To implement the PL/SQL procedures, you must not issue any commit or rollback command.

Because the connection used to call the security PL/SQL procedures is the connection used to persist all the metadata, this connection is controlled by Warehouse Builder through its own transaction manager to determine when to commit or rollback.

If you invoke commit or rollback in the procedure, it will corrupt the repository. If you want to record debug information in your PL/SQL procedure, you can do it in a separate child procedure, with autonomous transactions (use `pragma autonomous_transaction` at the beginning of the procedure). In this procedure, you can issue a commit or rollback command.

- 1)
- If you grant a user the permission to `CREATE` an object, the user also needs permission to `EDIT` the object. After an object is created, you can change your policy to revoke the user's `EDIT` permission on the object.
 - Do not perform any DML operations on the repository database objects created by Warehouse Builder when you implement the security PL/SQL procedures. This may damage the repository and make the Warehouse Builder client malfunction or stop functioning.
-

[Go to previous
page](#)

[Go to next
page](#)

[Oracle](#)
[Copyright © 1996, 2003 Oracle Corporation.](#)
All Rights Reserved.

[Go To Table
Of Contents](#)
[Contents](#)

[Go To
Index](#)
[Index](#)